

|     |     |   |
|-----|-----|---|
| -   | 42  | Macro definitions   |
| -   | 64  | Set up connect request services                           |
| 8-  | 111 | Accept incoming connect request                           |
| 9-  | 152 | Outgoing connect request                                  |
| 10- | 231 | Build session control portion of connect initiate message |
| 11- | 287 | Build process/task descriptor                             |
| 12- | 330 | Build image mode field                                    |
| 13- | 360 | Move image mode field                                     |
| 14- | 386 | Map node name to node address                             |
| 15- | 509 | Check for node name match                                 |
| 16- | 533 | Disconnect logical link                                   |
| 17- | 558 | Process received interrupt message                        |
| 18- | 581 | Complete interrupt message processing                     |

|     |        |        |        |            |                 |  |  |
|-----|--------|--------|--------|------------|-----------------|--|--|
| 500 | 001232 | 001006 |        | BNE        | 100\$           |  | ; If NE, an area was specified         |
| 501 | 001234 | 016400 | 000014 | MOV        | D\$NUM(R4),R0   |  | ; Supply our own area for this request |
| 502 | 001240 | 042700 | 001777 | BIC        | #*C<NM\$ARA>,R0 |  |  |
| 503 | 001244 | 050065 | 000012 | BIS        | R0,N\$NOD(R5)   |  |  |
| 504 |        |        |        |            |                 |  |  |
| 505 | 001250 | 005727 |        | 100\$: TST | (PC)+           |  | ; Valid node name mapping found        |
| 506 | 001252 | 000261 |        | 110\$: SEC |                 |  | ; Unable to find node name mapping     |
| 507 | 001254 |        |        | RETURN     |                 |  |  |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE      | REFERENCES                                       |
|----------|------------|--|
| W.MBOX   | 000012     | 17-576   |
| \$ACCT   | = ***** GX | 10-275   |
| \$CALLX  | = ***** GX | 8-131 14-457 14-474                              |
| \$SDSDFM | = ***** GX | 8-145 9-203 9-223 17-574 18-600 18-602 18-604    |
| \$DSNOD  | = ***** GX | 10-256 14-449 14-481 15-526 15-528 15-530 19-625 |
| \$FLAGS  | = ***** GX | 9-180 *14-405 *14-497                            |
| \$FLOW   | = ***** GX | 7-96   |
| \$IOPKT  | = ***** GX | 14-428   |
| \$LTM    | = ***** GX | 7-78   |
| \$MAIBX  | = ***** GX | *17-576  |
| \$PASSW  | = ***** GX | 10-270 14-456 14-467                             |
| \$REASN  | = ***** GX | 16-549   |
| \$REQID  | = ***** GX | 10-265 14-455 14-460                             |
| \$SRDFM  | = ***** GX | 10-260   |
| \$SRVCS  | = ***** GX | *7-108 8-139 9-214                               |

```

232 .SBTTL Transmit complete processing
233
234 *
235 **--XMTCP-Transmit complete processing
236 This routine is entered from the COMM/EXEC when a transmit has been
237 completed by the DLC layer.
238
239 -
240 Inputs:
241 R4 = Address of the CCB
242 C.NSP - Address of the associated I/O packet
243 C.STS - Completion status
244 C.CNT - Number of bytes transferred
245
246 Registers modified:
247 R0, R1, R2, R3, R4, R5
248
249 XMTCP: CALL STLNK ; Get address of physical link access block
250 DECBL $TIP(R5) ; Decrement count of transmits in progress
251 MOV C.STS(R4),R0 ; Get completion status
252 MOV C.CNT(R4),R1 ; Get # of bytes transferred
253 MOV C.NSP(R4),R3 ; Get address of the I/O packet
254
255 CMP R0,#IS.SUC ; Successful completion?
256 BEQ 5$ ; BR if yes
257 MOV R0,R1 ; Save error code
258 MOV #IE.VER,R0 ; And indicate error
259 CMPB #CB.RDB,C.BID(R4)
260 BNE 10$ ; If NE, this must be the big buffer
261 CALL RLSCN ; Release the buffer chain
262 BR 20$ ; and enter common code
263
264 10$: CLRB $BUFFG ; Mark the big buffer free
265 CALL @CCBRT ; and release the CCB
266
267 20$: CALLR RQDON ; Complete the I/O request
268

```



DLXCEX      CREATED BY    MACRO    ON 28-JUN-85 AT 22:53      PAGE 2      B 5

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES                         |
|---------|------------|------------------------------------|
| IS\$AS  | = *****    | 5-48                               |
| I.FCN   | = ***** GX | 19-504 19-528                      |
| I.LN2   | = ***** GX | 19-519                             |
| I.PRM   | = ***** GX | 13-288 13-289 13-290 13-318 13-323 |
| I.UCB   | = ***** GX | 19-506                             |
| KISAR6  | = ***** GX | *13-318                            |
| LENB    | = ***** GX | 9-116 10-174                       |
| LF\$BRO | = 002000   | 17-442 17-451 19-526               |
| LF\$CHN | = 004000   | 17-446                             |
| LF\$ENB | = 000400   | 9-115 9-119 19-540 19-541          |
| LF\$MOP | = 001000   | 13-329                             |
| LF\$RSR | = 100000   | 11-196                             |
| LF.ACT  | = 100000   | #5-50                              |
| LF.BRO  | = 000400   | #5-50                              |
| LF.BWT  | = 000007   | #5-50                              |
| LF.ENA  | = 002000   | #5-50                              |
| LF.LPB  | = 001000   | #5-50                              |
| LF.MDC  | = 000100   | #5-50                              |
| LF.MF   | = 004000   | #5-50                              |
| LF.MTP  | = 000020   | #5-50                              |
| LF.PAC  | = 000200   | #5-50                              |
| LF.RDY  | = 040000   | #5-50                              |
| LF.REA  | = 010000   | #5-50                              |
| LF.SER  | = 000040   | #5-50                              |
| LF.TIM  | = 000010   | #5-50                              |
| LF.UNL  | = 020000   | #5-50                              |
| LF.X2P  | = 000000   | #5-50                              |
| LLCRS   | = ***** GX | 9-140                              |
| LN.CLO  | = 000000   | #5-50                              |
| LN.DUM  | = 000005   | #5-50                              |
| LN.LOA  | = 000004   | #5-50                              |
| LN.LOO  | = 000003   | #5-50                              |
| LN.OAU  | = 000003   | #5-50                              |
| LN.OFF  | = 000001   | #5-50 10-163 10-166                |
| LN.ON   | = 000000   | #5-50                              |
| LN.OOP  | = 000004   | #5-50                              |
| LN.OPE  | = 000001   | #5-50                              |
| LN.REF  | = 000002   | #5-50                              |
| LN.SER  | = 000002   | #5-50                              |
| LN.STA  | = 000017   | #5-50                              |
| LN.SUB  | = 000360   | #5-50                              |
| LN.TRI  | = 000006   | #5-50                              |
| LR\$DEA | = 000200   | 19-528                             |
| LR\$DIS | = 000020   | 10-176                             |
| LR\$STP | = 000010   | 10-173                             |
| LS\$DAT | = 000004   | 17-456                             |
| LS\$ERR | = 000200   | 13-332                             |
| LS\$IDL | = 000001   | 16-421 19-498                      |
| LS\$MSK | = 000037   | 11-197                             |
| LS\$RST | = 000020   | 19-499                             |
| LS\$STR | = 000002   | 17-439                             |
| LS\$CTL | = 000012   | 10-169 *10-171 19-536 *19-538      |

```

265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280 000400 026027 000002 000006 XMTADR: CMP C.DATI(R0),#6 ; Are there at least 6 bytes of address?
281 000406 103416 BLO 10$ ; If L0, no
282
283 000410 016067 000010 000000G MOV C.CHRL(R0), $ADR ; Fill in Ethernet address
284 000416 016067 000012 000002G MOV C.CHRL+2(R0), $ADR+2
285 000424 016067 000014 000004G MOV C.CHRL+4(R0), $ADR+4
286 000432 052767 000000G 000000G BIS #FL$ADR, $FLAGS ; Indicate Ethernet address seen
287 000440 012701 000001 MOV #CS.SUC, R1 ; Return successful completion
288
289 000444 10$: RETURN

```

```

111 .SBTTL Control QIO function processing
112
113 *--NXCTL-Control QIO function processing
114
115 This function is called to perform network management control functions
116 for a physical link.
117
118 Inputs:
119 R3 = Address of I/O packet
120 I.PRM - Bias of <Line-id> string
121 I.PRM+2 - Virtual address of <Line-id> string
122 I.PRM+4 - Length of <Line-id> string
123
124 Registers modified:
125 R0, R1, R2, R3, R4, R5
126
127 000044 116304 000000G NXCTL:: MOVB I.FCN(R3),R4 ; Get subfunction code
128 000050 006204 ; ASR R4 ; Form table index
129 000052 006204 ; ASR R4 ; ...
130 000054 012700 000000C MOV #IE.ONP&377,R0 ; Assume invalid subfunction code
131 000060 022704 000022 CMP #SFMAX,R4 ; Check for valid code
132 000064 101545 BLOS CTLERR ; If LOS, error
133
134 000066 012702 177777 MOV #-1,R2 ; Indicate dynamic channel assignment not required
135 000072 CALL @SRCTBL(R4) ; Search for <Line-id> and validate user
136 000076 103540 BCS CTLERR ; If CS, error
137
138 .IF DF N$$BUF
139
140 000100 SAVRG <R0> ; Save PDV and Channel
141 000102 CALL @ISTBF ; Can this I/O request be buffered?
142 000105 103405 BCS 10$ ; If CS, no
143 000110 052763 100000 000000G BIS #100000,I.FCN(R3)
144 000116 CALL @INIBF ; Initiate buffering
145 000122 RESRG <R0> ; Recover PDV and Channel
146
147 .ENDC
148
149 000124 CALLR @FNCTBL(R4) ; Dispatch to processing routine

```

```

552                                     .SBTTL  Continue with deassign operation
553                                     ;+
554                                     ***-CTLDE1-Continue with deassign operation
555                                     ;
556                                     Update the COMM/EXEC databases to reflect the new line deassignment.
557                                     ;
558                                     -
559                                     Inputs:
560                                     R0 = PDV and Channel of target LLC
561                                     R1 = Address in target PDV channel mapping table
562                                     R2 = SLN and logical tributary # of <Line-id>
563                                     R3 = Address of the I/O packet
564                                     The N-bit is set from a 'TSTB C,STS(R4)
565 001106 100415 CTLDE1: BMI      CTLAB0      ; If MI, an error occurred
566                                     ;
567 001110                                     CALL    DEASSN      ; Modify the COMM/EXEC databases
568 001114 000421 BR        CTLDON      ; and complete the request

```

DLXDAT - DLX data bases MACRO V05.3b Friday 28-Jun-85 22:54 Page 5  
Macro definitions

44  
45  
46  
47  
48 000000

.SBTTL Macro definitions

.MCALL PLADF\$

PLADF\$ ; Define PLA offsets

```

82      .SBTTL  Stop operation of a physical channel
83      ;+
84      ;**--ABORQ--Abort operation of a physical channel
85      ;
86      ;      Stop operation of a physical channel by issuing a stop request to the
87      ;      channel.
88      ;
89      ;Inputs:
90      ;      R0 = Functions requested.(stop and/or disable)
91      ;      R5 = Address of the physical link access block
92      ;
93      ;Registers modified:
94      ;      R0, R1, R4
95      ;
96      ;      .ENABL  LSB
97
98      ABORQ:: MOV  B  #LS$STP,R1      ; Set new state to stopping
99              BIT   B  #LF$BRO,(R5)  ; Is this a broadcast channel?
100             BEQ   B  20$             ; If EQ, no
101             BIS   B  #LR$DIS,R0     ; Always force a disable function
102             BR    B  20$            ; Enter common code
103
104      ;+
105      ;**--DISRQ--Disable a physical channel
106      ;
107      ;      Stop operation of a physical channel by issuing a stop and a disable
108      ;      request to the channel.
109      ;
110      ;Inputs:
111      ;      R5 = Address of the physical link access block
112      ;
113      ;Registers modified:
114      ;      R0, R1, R4
115
116      DISRQ:: MOV  B  #LR$STP!LR$DIS,R0
117             BR    B  10$            ; Enter common code
118
119      ;+
120      ;**--STPRQ--Stop operation of a physical channel
121      ;
122      ;      Stop operation of a physical channel by issuing a stop request to the
123      ;      channel.
124      ;
125      ;Inputs:
126      ;      R5 = Address of the physical link access block
127      ;
128      ;Registers modified:
129      ;      R0, R1, R4
130
131      STPRQ:: MOV  B  #LR$STP,R0      ; Indicate stop required
132              MOV  B  #LS$ERR!LS$RST,R1
133
134              20$: BIT   B  #LF$ENB,(R5)  ; Is the link enabled?
135                  BNE   B  30$             ; If NE, yes
136                  BIC   B  #LR$DIS,R0     ; No need to issue the disable
137
138              30$: BITB  B  #LS$STP!LS$DAT,(R5)

```

70  
 71  
 72  
 73  
 74  
 75 000004

.SBTTL CONTEXT AREA OFFSET DEFINITIONS  
 ;  
 ; CONTEXT AREA OFFSET DEFINITIONS  
 ;

CICCX\$ LIST

; CONTEXT AREA OFFSET DEFINITIONS  
 ;

000000

000000  
 000020

.ASECT  
 = 0  
 L\$NLEN = 16. ; MAXIMUM NAME LENGTH

; The offsets L\$NAM to L\$PDV inclusive must appear in the same order  
 ; as in the CIRCX\$ and LOCCX\$ macros.  
 ;

000000  
 000000  
 000002  
 000003  
 000004  
 000020

L\$NAM: ; DEVICE NAME  
 L\$DDM: .BLKW 1 ; CONTROLLER NUMBER  
 L\$CTL: .BLKB 1 ; UNIT NUMBER  
 L\$UNT: .BLKB 1 ; 3RD WORD OF PVC NAME  
 L\$PVC: .BLKW 1 ; FOR EXPANDED CIRCUIT NAME  
 ; = 0+L\$NLEN ; SPACE FOR EXPANDED CIRCUIT NAME  
 L\$SCN: .BLKB L\$NLEN ; CURRENT NAME IN WILDCARD SCAN  
 L\$TRB: .BLKB 1 ; TRIBUTARY NUMBER  
 L\$PFG: .BLKB 1 ; PARSE FLAGS  
 L\$SLT: .BLKW 1 ; Current SLT/PVC address  
 L\$NXT: .BLKW 1 ; Pointer to next SLT/PVC address  
 L\$SNM: .BLKW 1 ; REMAINING NUMBER OF SYSTEM LINES  
 L\$TPT: .BLKW 1 ; Current tributary pointer (-1 for PSI)  
 L\$TNM: .BLKB 1 ; REMAINING NUMBER OF TRIBUTARIES  
 L\$CTB: .BLKB 1 ; CURRENT TRIBUTARY NUMBER  
 L\$CHN: .BLKB 1 ; Channel number / X.25 port number and  
 L\$PDV: .BLKB 1 ; ... PDV assigned  
 L\$MSG: .BLKW 1 ; ERROR MESSAGE STRING POINTER  
 L\$BUF: .BLKW 1 ; SAVED BUFFER POINTER  
 L\$OPT: .BLKB 1 ; SAVED OPTIONS BYTE  
 L\$TYP: .BLKB 1 ; CIRCUIT-ID FORMAT TYPE  
 L\$FLG: .BLKW 1 ; FLAG WORD (FOR COMMAND)  
 L\$FL1: .BLKW 1 ; FLAG WORD (FOR COMMAND)  
 L\$FL2: .BLKW 1 ; FLAG WORD (FOR CURRENT CIRCUIT)  
 L\$PAR: .BLKW 1 ; CURRENT PARAMETER TYPE  
 L\$SLEN: .BLKB 1 ; Significant length of circuit name  
 L\$PRU: ; Line protocol  
 L\$MTYP: .BLKB 1 ; Network management circuit type  
 L\$FLX: .BLKW 1 ; Flags word for X.25 circuit commands  
 L\$STA: .BLKB 1 ; CIRCUIT STATE  
 L\$COS: .BLKB 1 ; CIRCUIT COST  
 L\$OWN: .BLKB 1 ; CIRCUIT OWNER (PDV INDEX)  
 L\$TAD: .BLKB 1 ; TRIBUTARY ADDRESS  
 L\$ACT: .BLKB 1 ; MULTIPOINT ACTIVE RATIO  
 L\$DEA: .BLKB 1 ; MULTIPOINT DEAD RATIO  
 L\$DDT: .BLKW 1 ; DMP DEAD TIMER  
 L\$DLT: .BLKW 1 ; DMP DELAY TIMER  
 L\$PLT: .BLKW 1 ; DMP POLL TIMER  
 L\$BBT: .BLKW 1 ; DMP BABBLE TIMER  
 L\$NMT: .BLKW 1 ; DMP NORMAL TIMER  
 L\$XMT: .BLKW 1 ; DMP TRANSMIT TIMER

DLXQIO      CREATED BY    MACRO    ON 29-JUN-85 AT 12:23

PAGE 4      B 12

SYMBOL CROSS REFERENCE

CREF    04.00

| SYMBOL    | VALUE      | REFERENCES  |
|-----------|------------|---|
| \$SAVAL   | = ***** GX | 8-111   |
| \$TMEFN   | = ***** GX | 8-149   |
| \$TMLUN   | = ***** GX | 8-149   |
| \$\$\$ARG | = 000002   | 8-149    9-203    9-206   |
|           |            | #8-149    8-149    #8-149    8-149    #8-149    8-149    #8-149    8-149    8-149 |
|           |            | #8-149    8-149    8-149    8-149    8-149    8-149    8-149    8-149    8-149    |
|           |            | 8-149    8-149    #8-149    8-149    8-149    8-149    8-149    8-149    8-149    |
|           |            | 8-149    8-149    8-149    8-149    #8-149    8-149    8-149    8-149    8-149    |
|           |            | 8-149    8-149    8-149    8-149    8-149    8-149    8-149    8-149    8-149     |



DLXSUB - DLX subroutines  
Deassign line from an LLC

MACRO V05.03b Friday 28-Jun-85 22:56 Page 12

B.13

347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399

000576 004567 177724  
000602 012711 100000  
000606 005015  
  
000610  
000614 005046  
000616  
000622 105765 000014  
000626 001424  
  
000630 010204  
000632 000304  
000634 042704 177700  
000640 006304  
000642 006304  
000644 060504  
000646 142764 000200 000022  
  
000654 116504 000014  
000660 062705 000022  
000664 132715 000200  
000670 001006  
000672 022525  
000674  
  
000700 012766 100000 000004  
000706  
000712 042615  
000714

```
.SBTTL Deassign line from an LLC
**--DEASSN-Deassign line from an LLC

The COMM/EXEC databases are updated to reflect the new line deassignment
and the system line and/or tributary are marked inactive.

Inputs:
R1 = Address of channel entry in target PDV
R2 = SLN and logical tributary # of the <Line-id>

DEASSN::JSR R5,COASN ; Compute address in reverse mapping table
MOV #ZS.ASN,(R1) ; Mark channel free
CLR (R5) ; Mark SLN and logical tributary free

+
**--CLACT-Clear active for system line and/or tributary
Clear the active flag for the system line and/or the tributary.

Inputs:
R2 = SLN and logical tributary of the <Line-id>

Registers modified:
R5

CLACT:: CALL STSLT ; Compute address of the SLT
CLR -(SP) ; Assume line will remain active
SAVRG <R4,R5> ; Save some registers
TSTB L.NSTA(R5) ; Is this a multipoint master or broadcast?
BEQ 20$ ; If EQ, no

MOV R2,R4 ; Isolate tributary #
SWAB R4 ; ...
BIC #*C<??>,R4 ; ...
ASL R4 ; Form double word offset
ASL R4 ; ...
ADD R5,R4 ; Compute address in SLT
BICB #SF.ACT,L.MPF(R4)

MOVB L.NSTA(R5),R4 ; Get # of tributaries on the line
ADD #L.MPF,R5 ; Point to first tributary extension
10$: BITB #SF.ACT,(R5) ; Is this tributary still active?
BNE 30$ ; If NE, yes
CMP (R5)+,(R5)+ ; Skip over remainder of tributary entry
SOB R4,10$ ; Check all tributaries

30$: MOV #LF.ACT,4(SP) ; No active tributaries, bit to be cleared
RESRG <R5,R4> ; Restore registers
BIC (SP)+,(R5) ; Clear the status bit
RETURN
```

DLXSUB - DLX subroutines.  
Find an LLC process PDV index

MACRO V05.03b Friday 28-Jun-85 22:56 Page 20

```

885 .SBTTL Find an LLC process PDV index
886
887 **~FNDPR~Find an LLC process PDV index
888
889 Find the PDV index of an LLC process as supplied in the I/O packet.
890
891 Inputs:
892 R3 = Address of I/O packet
893 I.PRM+6 - RAD50 name of the LLC
894
895 Outputs:
896 R2 PDV index of LLC
897 'C' Clear - LLC process present and loaded
898 'C' Set - LLC process not present or not loaded
899
900 Registers modified:
901 R1
902
903 002254 016302 000006G FNDPR:: MOV I.PRM+6(R3),R2 ; Get LLC process name
904 002260 CALL @PDVID ; Find process PDV index
905 002264 103414 BCS 20$ ; If CS, process not in system
906
907 002266 010201 MOV R2,R1 ; Compute address in the
908 002270 067701 000000G ADD @PDVTA,R1 ; PDV index table
909 002274 011101 MOV (R1),R1 ; Get address of the PDV
910 002276 132761 000004 000010 BITB #ZF.LLC,Z.FLG(R1)
911 002304 001403 BEQ 10$ ; If EQ, process is not an LLC
912 002306 005761 000012 TST Z.PCB(R1) ; Is the process loaded?
913 002312 001001 BNE 20$ ; If NE, yes
914 002314 000261 10$: SEC ; Indicate process not present or not loaded
915 002316 20$: RETURN

```

DLXSUB      CREATED BY    MACRO    ON 28-JUN-85 AT 22:56      PAGE 4      B 15  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL   | VALUE      | REFERENCES  |
|----------|------------|---|
| ZF.X3P   | = 000000   | #5-48   |
| ZS.ASN   | = 100000   | #5-48      6-73      12-360   |
| ZS.BSY   | = 140000   | #5-48   |
| Z.AVL    | 000014     | #5-48   |
| Z.DAT    | 000016     | #5-48   |
| Z.DSP    | 000000     | #5-48      5-48   |
| Z.FLG    | 000010     | #5-48      18-725      19-847      20-910   |
| Z.LEN    | = 000016   | #5-48   |
| Z.LLN    | 000006     | #5-48      17-678   |
| Z.MAP    | 000020     | #5-48      17-675      17-678      21-938      22-966   |
| Z.NAM    | 000004     | #5-48   |
| Z.PCB    | 000012     | #5-48      20-912      23-990   |
| Z.SCH    | 000007     | #5-48   |
| \$ACCESS | = ***** GX | *16-599      18-777   |
| \$BIAS   | = ***** GX | *14-441      14-466   |
| \$BYTES  | = ***** GX | *14-443      14-460      14-462      14-463      *14-475  |
| \$DLXPD  | = ***** GX | 21-931  |
| \$PBLK   | = ***** GX | 15-511      *15-526      *15-530      *15-531      *15-532      16-603      16-605      16-607      16-622    |
|          |            | 17-662      17-664      17-667      17-669      18-718      *18-722      *18-726      18-738      18-740      |
|          |            | 18-742      *18-749      *18-750      *18-751      *18-752      *18-756      *18-758      18-762      *18-769 |
|          |            | *18-770      18-773      18-775      18-785      *18-789      *19-824      *19-825      *19-826      *19-827  |
|          |            | *19-838      *19-855      *19-865      *19-872      23-988  |
| \$VA     | = ***** GX | *14-442      14-467      *14-474  |

```
278 .SBTTL TIMER SERVICE
279
280 *--NETIME-TIMER SERVICE ENTRY POINT FROM THE COMM/EXEC
281
282 DURING TIMEOUT PROCESSING WE UNSTOP THE COLLECTION TASK IF THERE
283 ARE EVENTS AWAITING COLLECTION.
284
285 INPUTS:
286 R5 - POINTER TO EVENT LOGGER PROCESS' DATABASE
287
288 REGISTERS MODIFIED:
289 R0, R1, R2, R3
290
291 000426 005767 000076 NETIME: TST INIFLG ; HAVE WE INITIALISED BEFORE?
292 000432 001020 ; IF NE, YES
293
294 000434 012702 021274 MOV #*REVL,R2 ; FIND PDV INDEX OF EVENT LOGGER PROCESS
295 000440 @PDVID ;
296 000444 103413 BCS 10$ ; IF CS, FAILED TO FIND IT!!!
297
298 000446 010203 MOV R2,R3 ; COPY PDV INDEX
299 000450 067703 177356 ADD @PDVIA,R3 ; POINT INTO PDV INDFX TABLE
300 000454 011303 MOV (R3),R3 ; GET ADDRESS OF PDV
301 000456 016377 000016 177336 MOV Z,DAT(R3),@LGDDB ; SET UP ADDRESS OF DATA BASE
302 000464 010277 177334 MOV R2,@LGPDV ; SAVE PDV INDEX OF EVENT LOGGER PROCESS
303 000470 005267 000034 INC INIFLG ; SHOW PROCESS HAS BEEN INITIALISED
304
305 000474 016500 000004 10$: MOV E$TCB(R5),R0 ; GET COLLECTORS TCB/ATL ADDRESS
306 000500 001412 30$ BEQ 30$ ; IF EQ, NOT ACTIVE
307 000502 005765 000000 TST E$EVT(R5) ; ANY EVENTS AWAITING COLLECTION?
308 000506 001005 BNE 20$ ; IF NE, YES
309 000510 005777 177312 TST @LGSTT ; IS LOGGING CLOSING DOWN?
310 000514 001004 BNE 30$ ; IF NE, NO
311 000516 005065 000004 CLR E$TCB(R5) ; TELL COLLECTOR TO EXIT
312
313 000522 20$: CALL @EXRCN ; UNSTOP COLLECTOR TASK
314
315 000526 30$: RETURN
316
317 000530 000000 INIFLG: .WORD 0 ; INITIALISATION FLAG
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE SESUSR - Session control user interface routines
.IDENT /V05.00/
.ENABL LC
```

```
: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.
```

```
: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.
```

```
: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.
```

```
: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.
```

```
: Module description
```

```
: Session control user interface routines
```

```
: Ident history:
```

```
: 4.00 07-NOV-83
:       DECnet-11M V4.0
:       DECnet-11M-PLUS V2.0
:
: 5.00 22-JUL-85
:       DECnet-11M/S V4.2
:       DECnet-11M-Plus V3.0
:       DECnet-Micro/Rsx V1.0
:
```

```

509                                     .SBTTL  Check for node name match
510                                     +
511                                     **--CHKNAM-Check for node name match
512                                     :
513                                     Check if there is a match between an alias or remote name and the
514                                     node requested in an outgoing connect request.
515                                     -
516                                     Inputs:
517                                     R0 = Address of node name to match
518
519                                     Outputs:
520                                     'Z' Set - Node names match
521                                     'Z' Clear - Node names do not match
522
523                                     Registers modified:
524                                     R0
525
526 001256 022067 000000G CHKNAM: CMP      (R0)+,$DSNOD
527 001262 001005      BNE      10$
528 001264 022067 000002G      CMP      (R0)+,$DSNOD+2
529 001270 001002      BNE      10$
530 001272 022067 000004G      CMP      (R0)+,$DSNOD+4
531 001276      10$:      RETURN
  
```

## MACRO CROSS REFERENCE

CPEF 04.00

## MACRO NAME REFERENCES

|         |         |        |         |        |         |        |        |        |         |        |
|---------|---------|--------|---------|--------|---------|--------|--------|--------|---------|--------|
| ANBDF\$ | #6-46   | 6-57   |         |        |         |        |        |        |         |        |
| CALL    | 8-130   | 8-132  | 8-144   | 9-173  | 9-190   | 9-205  | 9-207  | 9-219  | 9-227   | 9-228  |
|         | 10-257  | 10-261 | 10-281  | 11-326 | 12-356  | 14-407 | 14-412 | 14-417 | 14-435  | 14-445 |
|         | 14-487  | 14-493 | 16-550  | 19-631 | 19-634  | 20-681 |        |        |         |        |
| CALLE   | #6-44   | 8-131  | 8-145   | 9-203  | 9-223   | 17-574 | 18-600 | 18-602 | 18-604  |        |
| CALLR   | 8-150   | 16-553 | 17-579  |        |         |        |        |        |         |        |
| CALLX   | #8-131  | 8-131  | #8-145  | 8- 45  | #9-203  | 9-203  | #9-223 | 9-223  | #17-574 | 17-574 |
|         | #18-600 | 18-600 | #18-602 | 18 502 | #18-604 | 18-604 |        |        |         |        |
| CCBDF\$ | #6-46   | 6-55   |         |        |         |        |        |        |         |        |
| COUNT\$ | #6-44   | 9-222  |         |        |         |        |        |        |         |        |
| CRBDF\$ | #6-45   | 6-50   |         |        |         |        |        |        |         |        |
| CTRDF\$ | #6-45   | 6-49   |         |        |         |        |        |        |         |        |
| DHBD\$  | #6-46   | 6-60   |         |        |         |        |        |        |         |        |
| ECDD\$  | #6-45   | 6-48   |         |        |         |        |        |        |         |        |
| LLTDF\$ | #6-45   | 6-51   |         |        |         |        |        |        |         |        |
| LLWDF\$ | #6-45   | 6-53   |         |        |         |        |        |        |         |        |
| MAP     | #6-44   | 8-129  |         |        |         |        |        |        |         |        |
| MAPLLT  | #6-44   | 8-129  |         |        |         |        |        |        |         |        |
| MBXDF\$ | #6-45   | 6-54   |         |        |         |        |        |        |         |        |
| MSGDF\$ | #6-45   | 6-52   |         |        |         |        |        |        |         |        |
| NSSYM\$ | #6-46   | 6-58   |         |        |         |        |        |        |         |        |
| RECMAP  | #6-46   | 14-406 |         |        |         |        |        |        |         |        |
| RESMAP  | #6-44   |        |         |        |         |        |        |        |         |        |
| RESRG   | #6-44   | 10-284 | 20-690  |        |         |        |        |        |         |        |
| RETURN  | 7-109   | 8-148  | 9-225   | 9-229  | 10-285  | 11-328 | 13-384 | 14-507 | 15-531  | 16-556 |
|         | 18-606  | 19-635 | 20-691  |        |         |        |        |        |         |        |
| RNBDF\$ | #6-46   | 6-56   |         |        |         |        |        |        |         |        |
| SAVMAP  | #6-44   |        |         |        |         |        |        |        |         |        |
| SAVRG   | #6-44   | 10-253 | 20-665  |        |         |        |        |        |         |        |
| SLIDF\$ | #6-46   | 6-59   |         |        |         |        |        |        |         |        |
| SOB     | 13-381  | 14-465 | 14-472  | 14-479 | 20-687  |        |        |        |         |        |

```

268 .SBTTL Receive complete processing
269
270 *
271 ***RCVCP-Receive complete processing
272
273 This routine is entered from the COMM/EXEC when a receive has been
274 completed by the DLC layer.
275
276 Inputs:
277 R4 = Address of the CCB (possibly first in a chain)
278 C.STS - Completion status
279 C.CNT - Number of bytes transferred
280
281 Registers modified:
282 R0, R1, R2, R3, R4, R5
283
284 RCVCP: CALL STLNK ; Get address of physical link access block
285 MOV L$RCV(R5),R3 ; Get next receive I/O packet
286 BEQ 50$ ; If EQ, none
287
288 MOV #IS.SUC&377,$IOS; Assume success
289 MOV I.PRM(R3),$BIAS ; Set up bias,
290 MOV I.PRM+2(R3),$VA ; virtual address and byte count
291 MOV I.PRM+4(R3),$BYTES
292
293 SAVRG <R3,R4,R5> ; Save I/O packet and CCB addresses
294
295 10$: MOV C.CNT(R4),R0 ; Get # of bytes in this buffer
296 CMP R0,$BYTES ; Is there enough space left in the buffer?
297 BLOS 20$ ; If LOS, yes
298 MOV $BYTES,R0 ; Use remaining byte count
299 MOV #IE.DA0&377,$IOS; and set data overrun status
300
301 20$: MOV (R4),-(SP) ; Save address of next CCB in the chain
302 TST R0 ; Any bytes to transfer?
303 BEQ 30$ ; If EQ, no
304 MOV C.BUF(R4),R1 ; Set source relocation bias
305 MOV C.BUF+2(R4),R2 ; and virtual address
306 SUB #20000,R2 ; biased for APR5
307
308 MOV $BIAS,R3 ; Set up destination bias
309 MOV $VA,R4 ; and the virtual address
310 ADD R0,$VA ; Update virtual address
311 SUB R0,$BYTES ; and byte count for next transfer
312 CALL @BLXIO ; Copy the data
313
314 30$: MOV (SP)+,R4 ; Get address of next CCB in the chain
315 BNE 10$ ; Loop till all data is copied
316
317 RESRG <R5,R4,R3> ; Recover CCB and I/O packet addresses
318
319 MAP I.PRM+6(R3) ; Map to the auxiliary characteristics buffer
320 BEQ 40$ ; If EQ, none present
321 CALL RCVCHP ; Process received message characteristics
322
323 40$: CALL RLSCHN ; Release the CCB chain
324 MOV I.PRM+4(R3),R1 ; Get size of user's buffer
325 SUB $BYTES,R1 ; Compute # of bytes transferred
  
```



DLXCEX      CREATED BY    MACRO    ON 28-JUN-85 AT 22:53      PAGE 3      C 5  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES   |
|---------|------------|--|
| L\$ERR  | 000010     | *13-331  |
| L\$FLG  | 000001     | 19-539   |
| L\$FNC  | 000014     | 17-448      18-475      19-502      19-544      *19-545    |
| L\$LEN  | 000026     | 11-216   |
| L\$RCV  | 000020     | 13-284   |
| L\$RSR  | 000002     | *14-360      19-494  |
| L\$STT  | 000007     | *10-166  |
| L\$TCB  | 000022     | *10-178      15-401      *19-520      *19-523      *19-533 |
| L\$TIM  | 000004     | 11-207      *11-209  |
| L\$TIP  | 000003     | *12-249  |
| L.COST  | 000015     | #5-50  |
| L.CTL   | 000012     | #5-50  |
| L.CVA   | 177776     | #5-50  |
| L.DDM   | 000002     | #5-50  |
| L.DDS   | 000004     | #5-50  |
| L.DLC   | 000003     | #5-50  |
| L.DLM   | 000006     | #5-50  |
| L.DLS   | 000010     | #5-50  |
| L.FLG   | 000000     | #5-50  |
| L.KRBA  | 000016     | #5-50  |
| L.LEN   | = 000022   | #5-50  |
| L.MPF   | 000022     | #5-50  |
| L.NMST  | 000020     | #5-50  |
| L.NSTA  | 000014     | #5-50  |
| L.OWNR  | 000021     | #5-50  |
| L.UNT   | 000013     | #5-50  |
| NULL    | 000336 R   | #11-221      11-227  |
| NXINA   | = ***** GX | 19-550   |
| N\$SVCT | = *****    | 13-318   |
| PLAST   | = ***** GX | 11-191   |
| PRO\$OV | = 000000   | #5-51  |
| RCVCHR  | = ***** GX | 13-320   |
| RCVCP   | = 000444 R | 7-73      #13-283  |
| RCVDN   | = ***** GX | 11-214      13-326   |
| RCVERY  | = 000340 R | 11-200      #11-226  |
| RLSCHN  | = ***** GX | 12-260      13-322      13-328                             |
| RQALT   | = ***** GX | 17-454   |
| RQDON   | = ***** GX | 12-266      18-477   |
| R\$SMPL | = *****    | 19-509   |
| R\$11D  | = *****    | 5-48   |
| R\$11M  | = 000000   | 5-48   |
| R\$11S  | = *****    | 5-48   |
| SF.ACT  | = 000200   | #5-50  |
| SF.ENA  | = 000100   | #5-50  |
| SF.LPB  | = 000004   | #5-50  |
| SF.MFL  | = 000040   | #5-50  |
| SF.PAC  | = 000020   | #5-50  |
| SF.REA  | = 000010   | #5-50  |
| SF.SER  | = 000001   | #5-50  |
| SF.SVC  | = 000002   | #5-50  |
| SF.UNL  | = 000040   | #5-50  |
| SHUT    | 000132 R   | 2-93      #10-160  |

```

291                                     .SBTTL Define Ethernet protocol for transmit
292                                     ;+
293                                     ;**--XMTPRO-Define Ethernet protocol for transmit
294                                     ;
295                                     ; Define the Ethernet protocol to be used for this message transmission.
296                                     ;
297                                     ; Inputs:
298                                     ; R0 = Address of characteristics block
299                                     ; R1 = Default status (CS.IGN = Characteristic ignored)
300                                     ; R3 = Address of the I/O packet
301                                     ; R5 = Address of the physical link access block
302                                     ;
303                                     ; Outputs:
304                                     ; R1 = Status to return for characteristic
305                                     ;
306 000446 026027 000002 000002 XMTPRO: CMP C.DATI(R0),#2 ; Are there at least 2 bytes of protocol?
307 000454 103410                                BLO 10$ ; If LO, no
308
309 000456 016067 000010 000000G MOV C.CHRL(R0),$PRO ; Fill in Ethernet protocol
310 000464 052767 000000G 000000G BIS #FL$PRO,$FLAGS ; Indicate Ethernet protocol seen
311 000472 012701 000001 MOV #CS.SUC,R1 ; Return successful completion
312
313 000476                                10$: RETURN
314
315                                000001 .END

```

```

151                                     .SBTTL  turn <Line-id> on
152
153                                     *
154                                     **--CTLON-Turn <Line-id> on
155                                     This routine is called to start the operation of the LLC process
156                                     assigned to the <Line-id>.
157
158                                     -
159                                     Inputs:
160                                     R1 = Address of system line table
161                                     R3 = Address of I/O packet
162
163                                     Registers modified:
164                                     R0, R1, R2, R3, R4, R5
165
166 000130 005767 000016G      CTLON:  IST    $PBLK+T$ASN      ; Is <Line-id> assigned to an LLC?
167 000134 001511              BEQ      IEALC          ; If EQ, no
168 000136              CALL    PRCLD          ; Is the target process loaded?
169 000142 103516              BCS      CTLERR       ; If CS, no
170 000144              CALL    @CCBG1         ; Allocate a CCB
171 000150 103511              BCS      IENOD       ; If CS, error
172
173 000152 016702 000010G      MOV      $PBLK+T$SLN,R2 ; Get SLN and logical tributary
174 000156              CALL    STACT          ; Mark system line (and tributary) active
175 000162 012764 000004 000012 MOV      #STRT,C.STS(R4) ; Indicate startup required
176 000170 000530              BR       CTLON1      ; Enter common code

```

```

570
571
572
573
574
575
576
577
578
579
580
581
582
583 001116 100411
584
585 001120
586 001124
587 001130 142715 000017
588 001134 152715 000001
589 001140 000407

.SBTTL Continue with off processing
;*
**CTLOF2-Continue with off processing
Complete the processing associated with turning a line off.
Inputs:
R0 = PDV and Channel of target LLC
R1 = Address in target PDV channel mapping table
R2 = SLN and logical tributary # of <Line-id>
R3 = Address of the I/O packet
The N-bit is set from a 'TSTB' C,STS(R4)
CTLOF2: BMI CTLABO ; If MI, an error has occurred
CALL CLACT ; Clear line/tributary active flag
CALL STNMST ; Compute address of network management state byte
BICB #LN.STA,(R5) ; Set network management state to 'off'
BISB #LN.OFF,(R5) ;
BR CTLDON ; Complete the request

```

```

50          .SBTTL Local data
51
52 000000    $PBLK:: .BLKB T$LEN    ; Parameter block
53          .EVEN
54 000024 000000 000000    $BIGBF:: .WORD 0,0    ; Double word address of the big buffer
55 000030 000000    $BIGSZ:: .WORD 0    ; Size of the largest buffer
56 000032 000    $BUFFG:: .BYTE 0    ; Buffer allocation flag
57 000033 000    $DLXCH:: .BYTE 0    ; # of channels supported
58 000034 000000    $DLXPD:: .WORD 0    ; PDV index of DLX process
59
60 000036    $ADR:: .BLKW 3    ; Holds Ethernet address for transmission
61 000044    $PRO:: .BLKW 1    ; Holds Ethernet protocol for transmission
62 000046 000000    $FLAGS:: .WORD 0    ; Control flags
63          000001    FL$ADR:=1    ; Ethernet address has been supplied
64          000002    FL$PRO:=2    ; Ethernet protocol has been supplied
65
66 000050 000000    $ACCESS:: .WORD 0    ; Access type for broadcast channels
67
68 000052 000000    $BIAS:: .WORD 0    ; Current address bias
69 000054 000000    $VA:: .WORD 0    ; Current virtual address
70 000056 000000    $BYTES:: .WORD 0    ; # of bytes left to transfer
71 000060 000000    $IOS:: .WORD 0    ; I/O status value
72
73          .EVEN
74
75          .IF DF R$MPL
76
77 $OLRCB:: .WORD 0    ; CCB used for reconfiguration
78 $OLRLN:: .WORD 0    ; # of SLN's remaining to scan
79 $OLRSL:: .WORD 0    ; Current SLT index table address
80 $OLRST:: .WORD 0    ; Current tributary flag address
81 $OLRCT:: .WORD 0    ; Count of remaining tributaries to scan
82 $OLRRT:: .WORD 0    ; Return address for completion
83 $OLRBK:: .BLKB T$LEN    ; Parameter block for reconfiguration
84          .EVEN
85
86          .ENDC

```

```

139 000122 001002          BNE 40$      ; If NE, no need to issue the stop
140 000124 042700 000010    BIC #LR$STP,R0
141
142 000130 110115          40$: MOVB R1,(R5) ; Set new physical link state
143 000132 150065 000002    BISB R0,L$RSR(R5) ; and set requested functions
144 000136 001442          BEQ 100$      ; If EQ, all done
145
146 000140          CALL RCVAB ; Abort any pending receives
147 000144 105765 000003    TSTB L$TIP(R5) ; Any transmits in progress?
148 000150 001002          BNE 50$      ; If NE, yes
149 000152          CALL XMTAB ; Abort any pending transmits
150 000156          50$:
151
152          ;+
153          **--STPCHN-Stop operation of a physical channel
154          ;
155          Stop the data link layer protocol of a physical channel.
156          ;--
157          Inputs:
158          R5 = Address of the physical link access block
159          ;
160          Registers modified:
161          R0, R4
162
163 000156 132765 000010 000002 STPCHN::BITB #LR$STP,L$RSR(R5)
164 000164 001410          BEQ 60$      ; If EQ, stop not required
165 000166 012700 002006    MOV #FC,CTL!FS.STP,R0
166 000172          CALL DOFNC ; Issue control function
167 000176 103421          BCS 70$      ; If CS, resource allocation failure
168 000200 142765 000010 000002    BICB #LR$STP,L$RSR(R5)
169
170 000206 132765 000020 000002 60$: BITB #LR$DIS,L$RSR(R5)
171 000214 001412          BEQ 70$      ; If EQ, disable not required
172 000216 012700 013006    MOV #FC,CTL!FS.DIS,R0
173 000222          CALL DOFNC ; Issue control function
174 000226 103405          BCS 70$      ; If CS, resource allocation failure
175 000230 142765 000020 000002    BICB #LR$DIS,L$RSR(R5)
176 000236 042715 000400          BIC #LF$ENB,(R5) ; Mark the physical link disabled
177
178 000242          70$: RETURN
179
180 000244          100$: CALLR STOPC ; Complete function processing
181
182          .DSABL LSB
  
```

```

000122      L$BSA: .BLKB 1      ; DMP ACTIVE BASE
000123      L$BSD: .BLKB 1      ; DMP DYING BASE
000124      L$BSI: .BLKB 1      ; DMP INACTIVE BASE
000125      L$INA: .BLKB 1      ; DMP ACTIVE INCREMENT
000126      L$IND: .BLKB 1      ; DMP DYING INCREMENT
000127      L$INI: .BLKB 1      ; DMP INACTIVE INCREMENT
000130      L$TH1: .BLKB 1      ; DMP DEAD THRESHOLD
000131      L$TH2: .BLKB 1      ; DMP DYING THRESHOLD
000132      L$TH3: .BLKB 1      ; DMP INACTIVE THRESHOLD
000133      L$MXB: .BLKB 1      ; DMP MAXIMUM BLOCKS
000134      L$NTL: .BLKW 13     ; NTL MESSAGE BUFFER
000166      L$SCR: .BLKW 15     ; SCRATCH BUFFER
000224      L$LCI: .BLKW 1      ; COUNTER TIMER
000226      L$HTM: .BLKW 1      ; HELLO TIMER
000230      L$SER: .BLKW 1      ; SERVICE
000230      L$LTM: .BLKW 1      ; LISTEN TIMER
000232      L$XCH: .BLKW 1      ; X25 Logical Channel Number (LCN)
000234      L$LMB: .BLKW 1      ; X25 Max Block (Line)
000234      L$CMB: .BLKW 1      ; X25 Max Data (Circuit)
000236      L$NUML: .BLKW 1      ; DLM Number length
000236      L$DTEL: .BLKW 1      ; X25 DTE Length (Circuit)
000240      L$NUM: .BLKW 1      ; DLM Number
000240      L$DTE: .BLKB 8     ; X25 DTE (Circuit)
000250      L$DTEP: .BLKW 1      ; X25 Pointer to DTE descriptor (Circuit)
000252      L$MWN: .BLKW 1      ; X25 Max Window (Line)
000252      L$MXW: .BLKB 1      ; X25 Max Window (Circuit)
000253      L$MRT: .BLKW 1      ; X25 Max Retransmits (Line)
000253      L$MXR: .BLKB 1      ; DLM Max Recalls (Circuit)
000254      L$HBT: .BLKW 1      ; X25 Holdback Timer (Line)
000256      L$NTI: .BLKW 1      ; X25 Retransmit Timer (Line)
000256      L$RET: .BLKW 1      ; DLM Recall Timer (Circuit)
000260      L$CUS: .BLKW 1      ; DLM Usage
000262      L$BLK: .BLKW 1      ; DLM blocking state
      .IF DF R$$PRO ;PRO/DECnet
      L$LTY: .BLKW 1      ; Loopback Type
      L$MDT: .BLKW 1      ; Modem Test
      .ENDC ; DF R$$PRO
      .EVEN

000264      L$LEN: ; LENGTH OF CONTEXT AREA
000004      .PSECT

; FLAGS WORD BIT DEFINITIONS (L$FLG)
;
000001      L$REA = 1      ; READ COUNTERS OPERATION
000002      L$ZER = 2      ; ZERO COUNTERS OPERATION
000004      L$SKP = 4      ; SKIP NEXT "FIND NEXT LINE" OPERATION.
; THIS IS USED TO FORCE AN EXTRA PASS
; FOR A MULTIPOINT LINE TO RETURN THE
; CONTROLLER COUNTERS AS WELL AS ALL
; OF THE TRIBUTARY COUNTERS.
; CONNECTED TO VERSION 2.0 NCP
000010      L$VR2 = 10     ; MULTIPLE ADJACENCY FLAG
040000      L$MLT = 40000  ; SEGMENTED RESPONSE IN PROGRESS
100000      L$SEG = 100000

; PARSE FLAG DEFINITIONS (L$PFG)
;
000001      L$UNT = 1      ; UNIT NUMBER FOUND

```

DLXQIO      CREATED BY    MACRO    ON 29-JUN-85 AT 12:23      PAGE 5      C 12

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES  |
|------------|---|
| ALUN\$S    | #5-60      9-203  |
| CALL       | #8-111      8-138      8-178      9-202   |
| CICCX\$    | #5-59      6-75   |
| DIR\$      | #8-149      8-149      #8-160      8-160      #9-203      9-203      #9-206      9-206                      |
| GLUN\$S    | #5-60      9-206  |
| MOV\$      | #8-149      8-149      8-149      8-149      #8-149      8-149      8-149      #8-149      8-149      8-149 |
|            | #9-203      9-203      9-203      #9-206      9-206      9-206  |
| MVB\$      | #8-149      8-149   |
| OFF\$      | #9-206  |
| QDPB\$S    | #8-149      8-149   |
| QIOW\$S    | #5-60      8-149  |
| QIO\$S     | #5-60   |
| RESRG      | #5-59      8-142      9-211   |
| RETURN     | #8-184      9-212   |
| RVP\$      | #8-149      8-149      8-149      8-149   |
| SAVRG      | #5-59      8-130      9-201   |
| WSIG\$S    | #5-60      8-160  |



DLXSUB - DLX subroutines  
Deassign physical link

C.13  
MACRO V05.03b Friday 28-Jun-85 22:56 Page 13

401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414 000716  
415 000722

```
.SBTTL Deassign physical link
;+
;**-DSLIN-Deassign physical link
;   The physical link associated with the physical link acces block is
;   deassigned and so has no associated LLC.
;-
;Inputs:
;   R5 = Address of the physical link access block
;Registers modified:
;   R1, R2, R3
DSLIN:: CALL    GTSLN      ; Get SLN and logical tributary #
        CALLR   DEASSN    ; Deassign the physical link
```

DLXSUB - DLX subroutines  
Get SLN and logical tributary #

MACRO V05.03b Friday 28-Jun-85 22:56 Page 21

C.14

```

917
918
919
920
921
922
923
924
925
926
927
928
929
930
931 002320 016701 000000G
932 002324 067701 000000G
933 002330 011101
934 002332 005002
935 002334 156502 000006
936 002340 006302
937 002342 060201
938 002344 017701 000020
939 002350 011102
940 002352

```

```

.SBTTL Get SLN and logical tributary #
+
**--GTSLN-Get SLN and logical tributary #
Get the SLN and logical tributary # associated with a specified
physical link access block.
-
Inputs:
R5 = Address of the physical link access block
Outputs:
R1 = Address of entry in process' channel mapping table
R2 = SLN and logical tributary #
GTSLN:: MOV $DLXPD,R1 ; Compute address of the channel
ADD @PDVTA,R1 ; mapping table for this process
MOV (R1),R1
CLR R2 ; Get the channel #
BISB L$LIN(R5),R2 ;
ASL R2 ; Form word offset
ADD R2,R1 ; Offset into PDV
ADD #Z.MAP,R1 ;
MOV (R1),R2 ; Get SLN and logical tributary #
RETURN

```

DLXSUB      CREATED BY    MACRO    ON 28-JUN-85 AT 22:56      PAGE 5      C 15

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |         |         |         |         |         |         |        |        |        |
|------------|------------|---------|---------|---------|---------|---------|---------|--------|--------|--------|
| CALL       | 6-91       | 6-107   | 7-155   | 7-157   | 7-162   | 8-203   | 9-259   | 11-341 | 12-374 | 13-414 |
|            | 14-447     | 14-472  | 14-486  | 16-600  | 18-719  | 19-836  | 19-843  | 19-853 | 19-863 | 19-870 |
|            | 19-876     | 20-904  | 23-989  | 24-1006 | 25-1035 | 26-1055 |         |        |        |        |
|            | 13-415     | 18-779  |         |         |         |         |         |        |        |        |
| CALLR      | #5-45      | 5-47    |         |         |         |         |         |        |        |        |
| CCBDF\$    | #5-45      | 5-51    |         |         |         |         |         |        |        |        |
| CHRDF\$    | #5-44      | 10-287  | 19-829  |         |         |         |         |        |        |        |
| MAP        | #5-45      | 5-48    |         |         |         |         |         |        |        |        |
| PDVDF\$    | #5-45      | 5-49    |         |         |         |         |         |        |        |        |
| PLADF\$    | #5-44      |         |         |         |         |         |         |        |        |        |
| RESMAP     | #5-44      | 6-104   | 8-205   | 8-217   | 9-260   | 9-268   | 10-306  | 11-343 | 12-397 | 14-473 |
| RESRG      | #5-44      | 14-479  | 14-488  | 17-698  | 19-882  | 22-968  | 26-1072 |        |        |        |
|            | 6-111      | 7-146   | 7-167   | 8-218   | 9-269   | 10-307  | 11-345  | 12-399 | 14-482 | 14-490 |
| RETURN     | 15-522     | 15-533  | 16-623  | 16-637  | 17-699  | 18-792  | 19-883  | 20-915 | 21-940 | 22-969 |
|            | 23-994     | 24-1009 | 25-1039 | 26-1073 | 27-1093 |         |         |        |        |        |
| SAVMAP     | #5-44      |         |         |         |         |         |         |        |        |        |
| SAVRG      | #5-44      | 6-90    | 8-186   | 8-200   | 9-244   | 9-257   | 10-285  | 11-325 | 12-377 | 14-439 |
|            | 14-458     | 14-465  | 17-660  | 19-816  | 22-956  | 26-1054 |         |        |        |        |
| SLTDF\$    | #5-45      | 5-50    |         |         |         |         |         |        |        |        |
| SOB        | 12-394     | 15-520  | 17-687  | 18-745  |         |         |         |        |        |        |

```

319          .SBTTL  ALLOCATE BUFFER FOR EVENT
320
321      ;+
322      **--ALLOC-ALLOCATE BUFFER FOR EVENT
323
324      IF THE EVENT COLLECTOR IS ACTIVE AND THE LAST EVENT QUEUED WAS NOT THE
325      SPECIAL 'EVENT LOST' RECORD, ATTEMPT TO ALLOCATE A BUFFER. IF THE
326      ALLOCATION FAILS, QUEUE THE 'EVENT LOST' RECORD.
327
328      INPUTS:
329      R5 - POINTER TO EVENT LOGGER PROCESS DATABASE
330
331      OUTPUTS:
332      R3 - ADDRESS OF ALLOCATED BUFFER
333      'C' CLEAR - BUFFER SUCCESSFULLY ALLOCATED
334      'C' SET - NO BUFFER AVAILABLE
335
336      000532 0265L 000002 000000G ALLOC: CMP     E$EVT5+2(R5),#$LST
337      000540 001434          BEQ     30$      ; IF EQ, WE HAVE JUST LOST AN EVENT
338
339      000542          INHIB$      ; INHIBIT INTERRUPTS
340
341      .IF DF R$$MPL
342      .IF NDF R$$PRO
343
344      BIT     #F2.MP,@FMSK2      :::IS THIS A MULTI-PROCESSOR
345      BEQ     5$      :::BR IF NO
346      CALL    @MPLCK      :::INTERLOCK COMM/EXEC RESOURCES
347
348      5$:
349      .ENDC
350      .ENDC
351
352      000554 016703 000000G      MOV     $LGLST,R3      :::GET BUFFER FROM LIST
353      000560 001006          BNE     10$      :::IF NE, SUCCESSFULLY ALLOCATED
354      000562 012703 000000G      MOV     #$LST,R3      :::IF NE, SUCCESSFULLY ALLOCATED
355      000566          CALL    @QUEUE      :::QUEUE AN 'EVENT LOST' RECORD
356      000572 005003          CLR     R3      :::TO THE COLLECTOR TASK
357      000574 000402          BR      R3      :::NO BUFFER ALLOCATED
358      000576 011367 000000G      10$: MOV     (R3),$LGLST      :::REMOVE BUFFER FROM FREE LIST
359      000602          20$:
360
361      .IF DF R$$MPL
362      .IF NDF R$$PRO
363
364      BIT     #F2.MP,@FMSK2      :::IS THIS A MULTI-PROCESSOR
365      BEQ     25$      :::BR IF NO
366      CALL    @C(SP)+      :::CO-ROUTINE RETURN TO UNLOCK RESOURCES
367
368      25$:
369      .ENDC
370      .ENDC
371
372      000602          ENAB$      :::ENABLE INTERRUPTS
373
374      000606 005703          TST     R3      ; DID WE GET A BUFFER?
375      000610 001011          BNE     40$      ; IF NE, YES
376
377      000612          SAVRG      <R0,R1>      ; SAVE SOME REGISTERS
378      000616 016500 000004      MOV     E$ICB(R5),R0      ; UNSTOP COLLECTOR TASK
379      000622          CALL    @EXRGN      ; ...

```

```

42                                     .SBTTL Macro definitions
43
44                                     .MCALL SAVRG,RESRG,MAP,COUNT$,CALLE,SAVMAP,RESMAP,MAPLLT
45                                     .MCALL ECDDBS$,CTRDFS$,CRBDFS$,LLTDFS$,MSGDFS$,LLWDFS$,MBXDFS$
46                                     .MCALL CCBDFS$,RNBDFS$,ANBDFS$,NSSYMS$,SLIDFS$,DHBDFF$,RECMAP
47
48 000000                                ECDDBS$                                ; Define ECL database offsets
49 000000                                CTRDFS$                                ; Define counter block offsets
50 000000                                CRBDFS$                                ; Define connect request block offsets
51 000000                                LLTDFS$                                ; Define LLI offsets
52 000000                                MSGDFS$                                ; Define message values
53 000000                                LLWDFS$                                ; Define window block offsets
54 000000                                MBXDFS$                                ; Define mailbox offsets
55 000000                                CCBDFS$                                ; Define CCB offsets
56 000000                                RNBDFS$                                ; Define remote name block offsets
57 000000                                ANBDFS$                                ; Define alias name block offsets
58 000000                                NSSYMS$                                ; Define NS: symbols
59 000000                                SLIDFS$                                ; Define SLI symbols
60 000000                                DHBDFF$                                ; Define DEC home block offsets
61
62 000001                                N$$SES = 1                                ; This module is part of session control

```

```

533                                     .SBTTL Disconnect logical link
534                                     ;+
535                                     ;**--SNDDIS-Disconnect logical link
536                                     ;
537                                     ;   The user has requested the logical link to be disconnected.
538                                     ;   Set the link sub states appropriately
539                                     ;-
540                                     ; Inputs:
541                                     ;   R3 = Virtual address of LLT
542                                     ;   R5 = Address of database descriptor
543                                     ;
544                                     ; Registers modified:
545                                     ;   R0, R1, R2, R4
546                                     ;
547 001300 122713 000012 SNDDIS::CMPB  #ST$DIP,(R3) ; Is the link already in disconnect state
548 001304 001414      BEQ  20$ ; If EQ, yes - set user state to done
549 001306 016763 000000G 000100      MOV  $REASN,L.DCR(R3); Store disconnect reason code
550 001314      CALL SAVOPT ; Save optional data in the LLT
551 001320 112713 000012      MOVB  #ST$DIP,(R3) ; Set logical link state to disconnecting
552 001324 012763 001004 000036      MOV  #<NS$SDI*400>+US$DSC,L.USTA(R3)
553 001332      CALLR TRMLNK ; Try to terminate the logical link
554
555 001336 112763 000004 000036 20$: MOVB  #US$DSC,L.USTA(R3) ; Set user state to done
556 001344      RETURN

```

```

DDDDDDDDDDDDDDDD    LLL                                XXXX          XXXX
DDDDDDDDDDDDDDDD    LLL                                XXXX          XXXX
DDDDDDDDDDDDDDDD    LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDD              DDD  LLL                                XXXX          XXXX
DDDDDDDDDDDDDDDD    LLL                                XXXX          XXXX
DDDDDDDDDDDDDDDD    LLL                                XXXX          XXXX
DDDDDDDDDDDDDDDD    LLL                                XXXX          XXXX

```

|     |        |        |                |       |        |                         |  |   |
|-----|--------|--------|----------------|-------|--------|-------------------------|--|---|
| 325 | 000650 | 016700 | 000000G        |       | MOV    | \$IOS,R0                |  | ; Get the I/O completion status             |
| 326 | 000654 |        |                |       | CALLR  | RCVDN                   |  | ; Post completion on the I/O packet         |
| 327 |        |        |                |       |        |                         |  |   |
| 328 | 000660 |        |                | 50\$: | CALL   | RLSCHN                  |  | ; Release the CCB chain                     |
| 329 | 000664 | 032715 | 001000         |       | BIT    | #LF\$MOP,(R5)           |  | ; Ignore overrun errors in maintenance mode |
| 330 | 000670 | 001005 |                |       | BNE    | 60\$                    |  |   |
| 331 | 000672 | 012765 | 000000C 000010 |       | MOV    | #IE,DA0&377,L\$ERR(R5); |  | ; Report loss on next I/O request           |
| 332 | 000700 | 152715 | 000200         |       | BISB   | #LS\$ERR,(R5)           |  | ; Force error report on next I/O request    |
| 333 | 000704 |        |                | 60\$: | RETURN |                         |  |   |



DLXCEX      CREATED BY    MACRO    ON 28-JUN-85 AT 22:53      PAGE 4      D 5

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES                               |
|--------|------------|--|
| START  | 000040 R   | 8-92      #9-112                         |
| STLNK  | = ***** GX | 8-88      12-248      13-283      14-357 |
| STNMST | = ***** GX | 9-121      10-162                        |
| STOPC  | 001162 RG  | 14-376      #19-493                      |
| STPCHN | = ***** GX | 11-229      11-230                       |
| STPRQ  | = ***** GX | 15-403                                   |
| STRCHN | = ***** GX | 11-226      16-423                       |
| STRTC  | 001054 R   | 14-375      14-378      #17-439          |
| S.COST | 000001     | #5-50                                    |
| S.FLG  | 000000     | #5-50                                    |
| S.LEN  | 000004     | #5-50                                    |
| S.NMST | 000002     | #5-50                                    |
| S.OWNR | 000003     | #5-50                                    |
| TMOUT  | 000232 RG  | 7-71      #11-191                        |
| U.VCB  | = ***** GX | 19-507                                   |
| XF.ASS | = 000030   | #5-51                                    |
| XF.A   | = 000040   | #5-51                                    |
| XF.GMC | = 000010   | #5-51      5-51                          |
| XF.OFF | = 000010   | #5-51      5-51                          |
| XF.ON  | = 000000   | #5-51      5-51                          |
| XF.REA | = 000020   | #5-51                                    |
| XF.SMC | = 000000   | #5-51      5-51                          |
| XF.STD | = 000200   | #5-51      5-51                          |
| XF.STP | = 000010   | #5-51      5-51                          |
| XMTAB  | = ***** GX | 19-493                                   |
| XMTCP  | 000352 R   | 7-72      #12-248                        |
| XV.BRO | = 000001   | #5-51      17-453                        |
| XV.OTH | = 000000   | #5-51                                    |
| XSSMCB | = *****    | 5-48                                     |
| ZF.COU | = 001000   | #5-48                                    |
| ZF.DDM | = 000001   | #5-48                                    |
| ZF.DIA | = 004000   | #5-48                                    |
| ZF.DLC | = 000002   | #5-48                                    |
| ZF.DVP | = 100000   | #5-48                                    |
| ZF.INI | = 040000   | #5-48                                    |
| ZF.KMX | = 000020   | #5-48                                    |
| ZF.LLC | = 000004   | #5-48                                    |
| ZF.LMC | = 000100   | #5-48                                    |
| ZF.MAN | = 020000   | #5-48                                    |
| ZF.MFL | = 000010   | #5-48                                    |
| ZF.MTM | = 000400   | #5-48                                    |
| ZF.MUX | = 000040   | #5-48                                    |
| ZF.PSE | = 002000   | #5-48                                    |
| ZF.SLI | = 010000   | #5-48                                    |
| ZF.TIM | = 000200   | #5-48                                    |
| ZF.XSP | = 000000   | #5-48                                    |
| ZS.ASN | = 100000   | #5-48                                    |
| ZS.BSY | = 140000   | #5-48                                    |
| Z.AVL  | 000014     | #5-48                                    |
| Z.DAT  | 000016     | #5-48                                    |
| Z.DSP  | 000000     | #5-48      5-48                          |
| Z.FLG  | 000010     | #5-48                                    |

|                |                |                 |                   |                  |
|----------------|----------------|-----------------|-------------------|------------------|
| ASSCHK= 000000 | CE.UDF= 100001 | CS.LST= 040000  | FC.CTL= 000006    | I.PRM = ***** GX |
| ASSCPS= 000000 | CE.UNS= 100344 | CS.MTL= 004000  | FC.KCP= 000016    | KISAR6= ***** GX |
| ASSPRI= 000000 | CF.CHN= 000001 | CS.RNG= 000010  | FC.KIL= 000004    | KSSCNT= 177546   |
| ASSTRP= 000000 | CF.EOM= 000004 | CS.ROV= 000004  | FC.MAN= 000024    | KSSCSR= 177546   |
| CB.CCB= 000002 | CF.HDR= 000020 | CS.RSN= 010000  | FC.MLD= 000026    | KSSDC= 000000    |
| CB.BDM= 000940 | CF.LB = 100000 | CS.SHU= 000001  | FC.PCT= 000030    | KSSTPS= 000074   |
| CB.DLC= 000020 | CF.LIN= 000002 | CS.SID= 000002  | FC.PWR= 000022    | LD\$LP = 000000  |
| CB.RDB= 000004 | CF.SOM= 000010 | CS.STR= 000004  | FC.RCE= 000002    | LF\$BRO= 002000  |
| CB.SDB= 000010 | CF.SYN= 000040 | CS.SUC= 000001  | FC.RCP= 000014    | LF\$CHN= 004000  |
| CB.SLI= 000100 | CF.TRN= 000100 | CS.TMO= 020000  | FC.TIM= 000010    | LF\$ENB= 000400  |
| CB.XLB= 000001 | CM.CIR= 000002 | CS.XUR= 000004  | FC.XCP= 000012    | LF\$MOP= 001000  |
| CC.ADR= 000100 | CM.DM1= 000253 | CS\$CKP= 000000 | FC.XME= 000000    | LF\$RSR= 100000  |
| CC.DAD= 000102 | CM.DM2= 000400 | CS\$ORE= 000400 | FL\$ADR= ***** GX | LR\$CTL= 000003  |
| CC.DPA= 000302 | CM.DM3= 000000 | CS\$RSH= 177564 | FL\$PRO= ***** GX | LR\$DEA= 000200  |
| CC.DST= 000200 | CM.FMT= 100000 | C.ADD 000034    | FS.AST= 000000    | LR\$DIS= 000020  |
| CC.ECM= 000311 | CM.HRD= 000002 | C.BID 000003    | FS.CIB= 002000    | LR\$STP= 000010  |
| CC.ECS= 000310 | CM.LIN= 000000 | C.BUF 000014    | FS.CRA= 001000    | LS\$DAT= 000004  |
| CC.ELS= 000304 | CM.LOO= 000001 | C.BUF1 000014   | FS.DIS= 013000    | LS\$ERR= 000200  |
| CC.FMC= 000300 | CM.LO1= 000317 | C.BUF2 000024   | FS.DVC= 001000    | LS\$FRE= 000000  |
| CC.EPH= 000303 | CM.LO2= 000000 | C.CHRL 000010   | FS.ENB= 012000    | LS\$IDL= 000001  |
| CC.EPR= 000301 | CM.LO3= 000000 | C.CNT 000020    | FS.EXI= 001000    | LS\$MSK= 000037  |
| CC.LCR= 000004 | CM.LT1= 000011 | C.CNT1 000020   | FS.GET= 006000    | LS\$RST= 000020  |
| CC.LDM= 000306 | CM.LT2= 000053 | C.CNT2 000030   | FS.HLT= 000000    | LS\$STP= 000010  |
| CC.LLC= 000200 | CM.LT3= 007400 | C.DAT1 000002   | FS.INI= 000000    | LS\$STR= 000002  |
| CC.LSA= 000307 | CM.RC1= 000253 | C.DATO 000004   | FS.KIL= 000000    | L\$CTL 000012    |
| CC.MCT= 000201 | CM.RC2= 001000 | C.FLG 000022    | FS.LCL= 100000    | L\$ERR 000010    |
| CC.NET= 000000 | CM.RC3= 000000 | C.FLG1 000022   | FS.LTM= 001000    | L\$FLG 000001    |
| CC.PDC= 000006 | CM.XLO= 000004 | C.FLG2 000032   | FS.MNT= 004000    | L\$FNC 000014    |
| CC.PRO= 000101 | CP.CON= 001140 | C.FNC 000010    | FS.MSN= 014000    | L\$LEN 000026    |
| CC.SAD= 000010 | CP.DCF= 000040 | C.LIN 000006    | FS.REA= 001000    | L\$LIN 000006    |
| CC.SID= 000305 | CP.DUM= 000540 | C.LNK 000000    | FS.RET= 000000    | L\$LST 000000    |
| CC.SPI= 000002 | CP.HDL= 000007 | C.MOD 000011    | FS.REZ= 003000    | L\$LUN 000024    |
| CC.STM= 000312 | CP.LAT= 002140 | C.NSP 000004    | FS.RLB= 002000    | L\$RCV 000020    |
| CE.ABO= 100362 | CP.LOO= 000220 | C.PRO 000042    | FS.RNG= 011000    | L\$RSR 000002    |
| CE.ACN= 100012 | CP.PS = 177400 | C.RSV 000002    | FS.RST= 000000    | L\$STT 000007    |
| CE.DAO= 100346 | CP.PS1= 000200 | C.STA 000007    | FS.RTN= 001000    | L\$TCB 000022    |
| CE.DIS= 100366 | CP.ROU= 001540 | C.STAT 000006   | FS.SET= 005000    | L\$TIM 000004    |
| CE.DNF= 100005 | CP.XCF= 000100 | C.STS 000012    | FS.SFC= 005000    | L\$TIM1 000005   |
| CE.ERR= 100370 | CP.2FR= 000030 | C.TYP 000000    | FS.SFR= 006000    | L\$TIP 000003    |
| CE.JLC= 100006 | CS.ABO= 000100 | C.URM 177776    | FS.SFS= 004000    | L\$XMT 000016    |
| CE.JLN= 100350 | CS.BRO= 000002 | C.XACP 000004   | FS.SPW= 040000    | L\$XASG= 000000  |
| CE.JNV= 100002 | CS.EJF= 000200 | C.XID 000035    | FS.STM= 000000    | L\$XDRV= 000000  |
| CE.JUM= 100013 | CS.CES= 000002 | C.XLEN 000044   | FS.STP= 002000    | L\$SP11= 000001  |
| CE.LTO= 100356 | CS.CHN= 000010 | C.XPLI 000000   | FS.STR= 001000    | L\$S11R= 000000  |
| CE.MCE= 100007 | CS.CMP= 000200 | C.XPT 00        | FS.TRM= 003000    | M\$SCRB= 000124  |
| CE.MOP= 100372 | CS.DAO= 000003 | C.XSVC 00       | FS.WLB= 001000    | M\$SCRX= 000000  |
| CE.NMA= 100014 | CS.DCR= 000400 | C.XTC 00        | FS.XKL= 002000    | M\$SFCS= 000000  |
| CE.NTE= 100361 | CS.DEF= 000004 | C.X25 0000      | FS.XOF= 010000    | M\$SMGE= 000000  |
| CE.PCN= 100011 | CS.DEV= 000002 | D\$SBUG= 177514 | FS.XCN= 007000    | M\$SNET= 000000  |
| CE.RFS= 100010 | CS.DIS= 000040 | D\$SISR= 000000 | FS.ZER= 002000    | M\$SOVR= 000000  |
| CE.RTE= 100376 | CS.ENA= 000001 | D\$SL11= 000000 | F\$SLVL= 000001   | N\$SACC= 000001  |
| CE.RTL= 100003 | CS.ENB= 000020 | D\$SYNC= 000000 | G\$STPP= 000000   | N\$SBUF= 000001  |
| CE.RTS= 100004 | CS.ERR= 100000 | D\$SYNM= 000000 | G\$STSS= 000000   | N\$SLDV= 000001  |
| CE.SRC= 100364 | CS.FTL= 001000 | D.BIAS 000000   | G\$STTK= 000000   | N\$SMCP= 000001  |
| CE.STP= 100352 | CS.HCR= 000001 | D.SIZE 000004   | G\$SWRD= 000000   | N\$SMML= 000001  |
| CE.TME= 100354 | CS.HFE= 002000 | E\$XPR= 000000  | I\$SRAR= 000000   | N\$SMOV= 000010  |
| CE.TMO= 100374 | CS.IGN= 000002 | FC.LCP= 000020  | I\$SPDN= 000000   | N\$SNCT= 000001  |

```

178 .SBTTL Reassign <Line-id> for service access
179
180 *--CTLSER-Reassign <Line-id> for service access
181
182 This routine is called by the network management service tasks to
183 change the <Line-id> and process assignment without forcing the
184 line to be disabled.
185
186 -
187 Inputs:
188 R1 = Address of system line table
189 R3 = Address of I/O packet
190 I.PRM+6 - RAD50 name of the target LLC
191 I.PRM+10- Privilege access flag:
192 0 - Open line non-privileged
193 <>0 - Open line privileged
194 Kill current user (if any)
195
196 Registers modified:
197 R0, R1, R2, R3, R4, R5
198
199 000172 016702 000022G CTLSER: MOV $PBLK+T$NMST,R2 ; Isolate current state of the line
200 000176 042702 177760 BIC #^C<LN.STA>,R2 ; ...
201 000202 020227 000002 CMP R2,#LN.SER ; Is the line already in service?
202 000206 001453 BEQ ISALT ; If EQ, yes
203 000210 020227 000003 CMP R2,#LN.OAU ; Autoservice is the same
204 000214 001450 BEQ ISALT ; ...
205 000216 020227 000000 CMP R2,#LN.ON ; Is the line on?
206 000222 001050 BNE IEABO ; If EQ, line in wrong state
207
208 000224 005763 000010G TST I.PRM+10(R3) ; Is this a privileged access?
209 000230 001450 BEQ IEWLK ; If EQ, no ... access denied
210
211 *--CTLREA-Reassign <Line-id> to another LLC
212
213 This routine is called to change the <Line-id> and process assignment
214 without forcing the line to be disabled.
215
216 -
217 Inputs:
218 R1 = Address of system line table
219 R3 = Address of I/O packet
220 I.PRM+6 - RAD50 name of the target LLC
221
222 Registers modified:
223 R0, R1, R2, R3, R4, R5
224
225 000232 CTLREA: CALL FNDPR ; Find the target process
226 000236 103460 BCS CTLERR ; If CS, no such process
227
228 000240 126702 000017G CMPB $PBLK+T$ASN+1,R2 ; Is <Line-id> already assigned to this process?
229 000244 001434 BEQ ISALT ; If EQ, yes ... all done
230 000246 CALL FNDPR ; Find a free channel on destination process
231 000252 103452 BCS CTLERR ; If CS, none available
232
233 000254 CALL @CCBGT ; Allocate a CCB
234 000260 103445 BCS IENOD ; If CS, allocation failure

```

```

591                                     .SBTTL Complete the control request
592                                     ;+
593                                     ; **CTLAB0-Complete the control request with an abort
594                                     ; **CTLRE2-Complete the control request with secondary status
595                                     ; **CTLDON-Complete the control request successfully
596                                     ;
597                                     ; Complete the control request.
598                                     ;
599                                     ; Inputs:
600                                     ; R3 = Address of the I/O packet
601                                     ; R4 = Address of the control CCB
602                                     ;
603                                     .ENABL LSB
604
605 001142 012700 000000C CTLAB0: MOV #IE.AB0&377,R0 ; Set abort status
606 001146 005001 CLR R1 ; No secondary status
607 001150 000406 BR 20$ ; Enter common code
608
609 001152 016401 000024 CTLRE2: MOV C.BUF2(R4),R1 ; Set up secondary status
610 001156 000401 BR 10$ ; Enter common code
611
612 001160 005001 CTLDON: CLR R1 ; No secondary status
613 001162 012700 000000G 10$: MOV #IS.SUC,R0 ; Set success status
614 001166 CALL @CCBRT ; Release the CCB
615
616 001172 CALLR RQDON ; Complete the I/O request
617
618 000001 .END

```

```

88          .SBTTL  Exec vector table
89
90          ;
91          ; Exec vector table
92
93          $DLXVT::
94          000062 000000          .WORD 0          ; Flags word
95          000064 000000G        KSAR6:: .WORD KISAR6
96          000066 000000G        ACHCK:: .WORD $ACHCK
97          000070 000000G        BLXIO:: .WORD $BLXIO
98          000072 000000G        INIBF:: .WORD $INIBF
99          000074 000000G        IOFIN:: .WORD $IOFIN
100
101          .IF DF  R$$MPL
102          NCPU:: .WORD $NCPU
103          .ENDC
104
105          000076 000000G        QUEBF:: .WORD $QUEBF
106          000100 000000G        RELOC:: .WORD $RELOC
107          .IF DF  R$$MPL
108          SCDEV:: .WORD $SCDEV
109          SCERR:: .WORD $SCERR
110          .ENDC
111          000102 000000G        TSTBF:: .WORD $TSTBF
112          .IF DF  R$$MPL
113          URMST:: .WORD $URMST
114          .ENDC
115          000104 000000G        CCBGT:: .WORD $CCBGT
116          000106 000000G        CCBRT:: .WORD $CCBRT
117          000110 000000G        CEMUL:: .WORD $CEMUL
118          000112 000000G        CMPDV:: .WORD $CMPDV
119          000114 000000G        LDBGT:: .WORD $LDBGT
120          000116 000000G        LLCRQ:: .WORD $LLCRQ
121          000120 000000G        LLCRS:: .WORD $LLCRS
122          000122 000000G        LLCTA:: .WORD $LLCTA
123          000124 000000G        MAXOV:: .WORD $MAXOV
124          000126 000000G        PDVID:: .WORD $PDVID
125          000130 000000G        PDVTA:: .WORD $PDVTA
126          000132 000000G        RDBRT:: .WORD $RDBRT
127          000134 000000G        RDBSZ:: .WORD $RDBSZ
128          000136 000000G        SLTMA:: .WORD $SLTMA
129          000140 000000G        SLTNM:: .WORD $SLTNM
130          000027          $DLXVL==<<.-$DLXVT>/2>-1          ; Vector table length

```

```

184                                     .SBTTL Start operation of a physical channel
185                                     :+
186                                     **--STRCHN-Start operation of a physical channel
187                                     :
188                                     Start the data link layer protocol of a physical channel.
189                                     :-
190                                     Inputs:
191                                     R5 = Address of physical link access block
192                                     :
193                                     Registers modified:
194                                     R0, R4
195
196 000250 042715 004000 STRCHN::BIC    #LF$CHN,(R5)    ; Assume chaining not supported
197 000254 032715 000400      BIT    #LF$ENB,(R5)    ; Is the physical link enabled?
198 000260 001010          BNE    10$              ; If NE, yes
199
200 000262 012700 012003      MOV    #FC.CTL!FS.ENB,R0
201 000266          CALL    DOFNC                    ; Issue control function
202 000272 103417          BCS    30$              ; If CS, resource allocation failure
203 000274 052715 000400      BIS    #LF$ENB,(R5)    ; Mark the physical link enabled
204 000300 000414          BR     30$              ; Wait for enable to complete
205
206 000302 012700 001006 10$: MOV    #FC.CTL!FS.STR,R0
207 000306 032715 001000      BIT    #LF$MOP,(R5)    ; Should we start the channel in maintenance mode?
208 000312 001402          BEQ    20$              ; If EQ, no
209 000314 012700 004006      MOV    #FC.CIL!FS.MNT,R0
210
211 000320          CALL    DOFNC                    ; Issue control function
212 000324 103402          BCS    30$              ; If CS, resource allocation failure
213 000326 112715 000002      MOVB   #LS$STR,(R5)    ; Set new state to 'Starting'
214
215 000332          30$: RETURN
216
217          000001          .END

```

```

000002      LP$TRB = 2      ; TRIBUTARY NUMBER FOUND
000004      LP$MUX = 4      ; DEVICE IS MUX
000010      LP$MPT = 10     ; LINE IS MULTIPOINT
000020      LP$WDV = 20     ; WILD CARD DEVICE NAME FOUND
000040      LP$WCN = 40     ; WILD CARD CONTROLLER NUMBER FOUND
000100      LP$WUN = 100    ; WILD CARD UNIT NUMBER FOUND
000200      LP$WTR = 200    ; WILD CARD TRIBUTARY NUMBER FOUND
000360      LP$WLD = LP$WDV!LP$WCN!LP$WUN!LP$WTR ; WILD CARD FIELD MASK

```

;; FLAGS WORD BIT DEFINITIONS (L\$FLG)

```

000001      L$NNTL=1      ; NTL SET FUNCTION
000002      L$NSTA=2      ; SET STATE
000004      L$SCOS=4      ; SET COST
000010      L$SOWN=10     ; SET OWNER
000020      L$STAD=20     ; SET TRIBUTARY ADDRESS
000040      L$SACT=40     ; SET MULTIPOINT ACTIVE
000100      L$SDEA=100    ; SET MULTIPOINT DEAD
000200      L$CNNTL=200   ; NTL CLEAR FUNCTION
000400      L$COWN=400    ; CLEAR OWNER
001000      L$SDDT=1000   ; SET DEAD TIMER
002000      L$SDLT=2000   ; SET DELAY TIMER
004000      L$SPLT=4000   ; SET POLL TIMER
010000      L$SBBT=10000  ; SET BABBLE TIMER
020000      L$SNMT=20000  ; SET NORMAL TIMER
040000      L$SXMT=40000  ; SET TRANSMIT TIMER
100000      L$SSER=100000 ; SET SERVICE [ENABLE/DISABLE]

```

;; FLAGS WORD BIT DEFINITIONS (L\$FL1)

```

000001      L$BSA=1      ; SET ACTIVE BASE
000002      L$BSD=2      ; SET DYING BASE
000004      L$BSI=4      ; SET INACTIVE BASE
000010      L$SINA=10    ; SET ACTIVE INCREMENT
000020      L$SIND=20    ; SET DYING INCREMENT
000040      L$SINI=40    ; SET INACTIVE INCREMENT
000100      L$STH1=100   ; SET DEAD THRESHOLD
000200      L$STH2=200   ; SET DYING THRESHOLD
000400      L$STH3=400   ; SET INACTIVE THRESHOLD
001000      L$SMB=1000   ; SET MAXIMUM BLOCKS
010000      L$SMTM=10000 ; SET HELLO TIMEP
020000      L$SMDT=20000 ; SET MODEM TEST
040000      L$SLOO=40000 ; SET CONTROLLER LOOPBACK
100000      L$SNOR=100000 ; SET CONTROLLER NORMAL

```

.IF DF R\$PRO ; PRO/DECNET

;; LOOPBACK TYPE WORD BIT DEFINITIONS (L\$LTYP)

```

L$INT = 0      ; INTERNAL LOOPBACK
L$EXT = 1      ; EXTERNAL LOOPBACK

```

;; MODEM TEST TYPE WORD BIT DEFINITIONS (L\$MDT)

```

L$LCCL = 0      ; LOCAL MODEM TEST
L$REM = 1      ; REMOTE MODEM TEST
L$OFF = 2      ; SHUT OFF MODEM TESTS
.ENDC ; DF R$PRO

```

\*\*FILE\*\*ID\*\*DLXSUB

```

DDDDDDDD LL      XX      XX      SSSSSSSS UU      UU      BBBB8888
DDDDDDDD LL      XX      XX      SSSSSSSS UU      UU      88888888
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SSSSSS UU      UU      88888888
DD      DD LL      XX      XX      SSSSSS UU      UU      88888888
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SS      UU      UU      88      88
DD      DD LL      XX      XX      SSSSSSSS UUUUUUUUUU 88888888
DDDDDDDD LLLLLLLLLL XX      XX      SSSSSSSS UUUUUUUUUU 88888888
DDDDDDDD LLLLLLLLLL XX      XX      SSSSSSSS UUUUUUUUUU 88888888

```

....  
....  
....  
....

```

LL      SSSSSSSS TTTTTTTTTT
LL      SSSSSSSS TTTTTTTTTT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LL      SSSSSS TT
LL      SSSSSS TT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```



DLXSUB - DLX subroutines MACRO V05.03b Friday 28-Jun-85 22:56 Page 14  
Allocate and fill a chain of large data buffers

```

417 .SBTTL Allocate and fill a chain of large data buffers
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439 000726
440
441 000730 016367 000000G 000000G MOV I.PRM(R3), $BIAS ; Set up bias,
442 000736 016367 000002G 000000G MOV I.PRM+2(R3), $VA ; virtual address and count
443 000744 016367 000004G 000000G MOV I.PRM+4(R3), $BYTES
444 000752 005046 CLR -(SP) ; Initialise chain listhead
445 000754 010601 MOV SP, R1 ; Save address of listhead
446
447 000756 10$: CALL @LDBGT ; Allocate a large data buffer
448 000762 103460 BCS 100$ ; If CS, allocation failure
449 000764 005716 TST (SP) ; Is this the first buffer in the chain?
450 000766 001406 BEQ 20$ ; If EQ, yes
451 000770 167764 000000G 000016 SUB @MAXOV, C.BUF+2(R4)
452 000776 067764 000000G 000020 ADD @MAXOV, C.CNT(R4); We can use the extra space in this buffer
453
454 001004 005014 20$: CLR (R4) ; Clear link
455 001006 019411 MOV R4, (R1) ; Add CCB to end of list
456 001010 010401 MOV R4, R1 ; Save end of list address
457
458 001012 SAVRG <R1> ; Save current end of list address
459 001014 016400 000020 MOV C.CNT(R4), R0 ; Get size of the buffer
460 001020 020067 000000G CMP R0, $BYTES ; Is there enough data left in the buffer?
461 001024 101405 BLOS 30$ ; If LOS, yes
462 001026 016700 000000G MOV $BYTES, R0 ; Use remaining buffer size
463 001032 016764 000000G 000020 MOV $BYTES, C.CNT(R4); and as the byte count in the CCB
464
465 001040 30$: SAVRG <R0> ; Save copy size
466 001042 016701 000000G MOV $BIAS, R1 ; Set up source relocation bias
467 001046 016702 000000G MOV $VA, R2 ; and virtual address
468 001052 162702 020000 SUB #20000, R2 ; biased for APR5
469
470 001056 016403 000014 MOV C.BUF(R4), R3 ; Set up destination relocation bias
471 001062 016404 000016 MOV C.BUF+2(R4), R4 ; and virtual address
472 001066 CALL @BLXIO ; Copy the data
473 001072 RESRG <R0, R1> ; Restore copy size and end of list address

```

```

942                                     .SBTTL  Compute address within process channel mapping table
943
944                                     ;+
945                                     **--MAPAD-Compute address within process channel mapping table
946                                     Compute the address of the channel mapping entry associated with a
947                                     given channel.
948                                     ; -
949                                     Inputs:
950                                     R0 = PDV and Channel assignment
951
952                                     Outputs:
953                                     R1 = Address of the process channel mapping table entry
954                                     R2 = Address of the PDV
955
956 002354 MAPAD:: SAVRG    <R5>          ; Get a free register
957 002356      MOV     R0,R5          ; Get the PDV and Channel
958 002360      MOV     R5,R1          ; Isolate PDV and Channel
959 002362      BIC     #377,R1        ; ...
960 002366      BIC     R1,R5          ; ...
961 002370      SWAB    R1            ; Compute address in PDV index table
962 002372      ADD     @PDVTA,R1      ; ...
963 002376      MOV     (R1),R1        ; Get address of the PDV
964 002400      MOV     R1,R2          ; Save address for output
965 002402      ASL     R5             ; Form word offset
966 002404      ADD     #Z.MAP,R1      ; Compute address of process channel
967 002410      ADD     R5,R1          ; mapping table entry
968 002412      RESRG   <R5>          ; Restore register
969 002414      RETURN
  
```

[illegible]

EVL - EVENT LOGGER PROCESS  
ALLOCATE BUFFER FOR EVENT

MACRO V05.03b Friday 28-Jun-85 22:58 Page 11-1

D 16

376 000626

RESRG <R1,R0>

; RESTORE REGISTERS

377

378 000632 000261

30\$:

SEC

; INDICATE ERROR

379 000634

40\$:

RETURN

```

64      .SBTTL Set up connect request services
65
66      ;+
67      ;**--SETREQ-Set up connect request services
68      ;
69      ;   Set up the connect request services that must be placed in the
70      ;   connect initiate or connect confirm message.
71      ;
72      ; Inputs:
73      ;   R3 = Virtual address of LLT
74      ;   R5 = Address of database descriptor
75      ;
76      ; Registers modified:
77      ;   R0
78 000000 016763 000000G 000066 SETREQ. MOV    $LTM,L.ILTT(R3) ; Set up long term timer initial value
79
80      .IF DF N$$SLI
81
82      MOVB  $FLOW,R0      ; Get requested flow control
83      BIT   #S$NTIF,R0    ; Does user require notification of flow control changes?
84      BEQ   10$           ; If EQ, no
85      BIS   #LT.NOT*400,(R3); Mark notification required
86
87      10$: BIT   #S$TMDA,R0 ; Does user require timer delayed ACK processing?
88      BEQ   20$           ; If EQ, no
89      BIS   #LT.TDA*400,(R3); Mark processing required
90
91      20$: BIC   #^C<LF,MMF!LF,MCF>,R0
92      BISB  R0,L.FLAG(R3) ; Set requested flow control
93
94      .IFF
95
96 000006 156763 000000G 000014 BISB  $FLOW,L.FLAG(R3); Set requested flow control
97
98      .ENDC
99
100 000014 012700 000001      MOV   #CL$TYP,R0      ; Set up services byte
101 000020 132763 000201 000014 BITB  #LF.MMF,L.FLAG(R3)
102 000026 001402      BEQ   30$           ; If EQ, not message flow control
103 000030 052700 000010      BIS   #CL$MFL,R0      ; Set message flow control
104 000034 132763 000100 000014 30$: BITB  #LF.MSF,L.FLAG(R3)
105 000042 001402      BEQ   40$           ; If EQ, not segment flow control
106 000044 052700 000004      BIS   #CL$SFL,R0      ; Set segment flow control
107
108 000050 110067 000000G 40$: MOVB  R0,$SRVCS      ; Set up services byte
109 000054      RETURN

```

```

558 .SBTTL Process received interrupt message
559
560 ;+
561 ;**-USRINT-Process received interrupt message
562 ;
563 ; Place a received interrupt message at the end of the task's
564 ; mailbox queue.
565 ;-
566 ; Inputs:
567 ; R4 = Address of CCB
568 ; C.STS - Physical address of LLT
569 ; R5 = Address of database descriptor
570 ;
571 ; Registers modified:
572 ; R0, R1, R2, R3, R4
573 ;
574 ; USRINT::MOV C.STS(R4),R3 ; Get physical address of LLT
575 ; CALLE ACCLLT ; Gain access to the LLT
576 ; MOV L.WIND(R3),R3 ; Get address of the window block
577 ; MOV W.MBOX(R3),%MAIBX
578 ; MOV #NT.INT,C.FNC(R4)
579 ; MOVB W.LUN(R3),C.MOD(R4)
580 ; CALLR ADDMAI ; Add item to mailbox
581
582 001346 016403 000012
583 001352
584 001362 016303 000040
585 001366 016367 000012 000000G
586 001374 012764 000002 000010
587 001402 116364 000003 000011
588 001410

```



```

335 .SBTTL Control complete processing
336
337 +
338 ***-CTLCP-Control complete processing
339 This routine is entered from the COMM/EXEC when a control complete
340 request is dequeued for this process.
341
342 -
343 Inputs:
344 R3 = Subfunction code
345 R4 = Address of control CCB
346
347 Outputs: (to processing routines)
348 R1 = Completion status
349 R2 = Error offset for characteristics functions
350 R5 = Address of physical link access block
351
352 Registers modified:
353 R0, R1, R2, R3, R4, R5
354
355 000706 005764 000006 CTLCP:: TST C,LIN(R4) ; Is completion from another LLC?
356 000712 100421 BMI 20$ ; If MI, yes
357
358 000714 CALL STLNK ; Get address of link control block
359 000720 MOV C,MOD(R4),R0 ; Recover subfunction code
360 000724 BEQ 10$ ; If EQ, asynchronous error
361 000726 DEC L$RSR(R5) ; Reduce count of active control requests
362
363 000732 016401 000012 10$: MOV C,STS(R4),R1 ; Get completion status
364 000736 016402 000016 MOV C,BUF1+2(R4),R2 ; Compute error offset for
365 000742 166402 000026 SUB C,BUF2+2(R4),R2 ; characteristics functions
366
367 000746 CALL @CCBRT ; Release the CCB
368 000752 CALLR @CTLTLBL(R0) ; Dispatch to processing routine
369
370 000756 20$: CALLR CTLCCP ; Process LLC control completions
371
372 +
373 -
374 Control complete dispatch table
375
376 CTLTLBL: .WORD ASYNCH ; Asynchronous error
377 .WORD STRIC ; Start complete
378 .WORD STOPC ; Stop complete
379 .WORD +1 ; Reserved
380 .WORD STRIC ; Maintenance start complete
381 .WORD CHARC ; Set characteristics complete
382 .WORD CHARC ; Get characteristics complete
383 .WORD +1 ; Reserved
384 .WORD +1
385 .WORD ENBLC ; Enable complete
386 .WORD STOPC ; Disable complete

```



DLXCEX      CREATED BY    MACRO    ON 28-JUN-85 AT 22:53      PAGE 5      E 5

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES                                       |
|---------|-------------|--|
| Z.LEN   | = 000016    | #5-48  |
| Z.LLN   | 000006      | #5-48  |
| Z.MAP   | 000020      | #5-48  |
| Z.NAM   | 000004      | #5-48  |
| Z.PCB   | 000012      | #5-48  |
| Z.SCH   | 000007      | #5-48  |
| \$BIAS  | = ***** GX  | *13-288    13-307                                |
| \$BUFFG | = ***** GX  | *12-263  |
| \$BYTES | = ***** GX  | *13-290    13-295    13-297    *13-310    13-324 |
| \$DLXCH | = ***** GX  | *11-192  |
| \$DLXTB | = 000000 RG | *7-67  |
| \$IOS   | = ***** GX  | *13-287    *13-298    13-325                     |
| \$VA    | = ***** GX  | *13-289    13-308    *13-309                     |

E 6

DLXCHR - DLX Auxilliary charact MACRO V05.03b Friday 28-Jun-85 22:53 Page 11-2  
Symbol table

|                 |                |                |                |                    |
|-----------------|----------------|----------------|----------------|--------------------|
| N\$PEM= 000001  | RCVTBL 000064R | T\$DDM 000014  | T\$PDV 000007  | XMTCHR 000314RG    |
| P\$P45= 000000  | R\$DER= 000000 | T\$DLC 000015  | T\$SFLG 000004 | XMTPRO 000446R     |
| P\$WRD= 000000  | R\$K11= 000001 | T\$FLG 000006  | T\$SLF 000012  | XMTTBL 000374R     |
| Q\$OPT= 000010  | R\$SND= 000000 | T\$KRB 000020  | T\$SLN 000010  | X\$DDBT= 000000    |
| RCMAX = 000006  | R\$11M= 000000 | T\$LEN 000024  | T\$KMG= 000000 | \$DR = ***** GX    |
| RCVADR 000072R  | S\$WRG= 000000 | T\$LIN 000003  | T\$MIN= 000000 | \$FLAGS= ***** GX  |
| RCVCHR 000000RG | S\$YSZ= 007600 | T\$NAM 000000  | V\$CTR= 001000 | \$PRO = ***** GX   |
| RCVDAD 000206R  | T\$ASN 000016  | T\$NMST 000022 | XMMAX = 000004 | .\$\$\$\$ = 000034 |
| RCVPRO 000146R  | T\$CTL 000002  | T\$NSTA 000005 | XMTADR 000400R |                    |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000500 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 11  
Work file writes: 14  
Size of work file: 17569 Words ( 69 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:15.26  
SY:DLXCHR.V2,[131,134]DLXCHR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]DLXCHR

DLXCTL - DLX control QIO functi MACRO V05.03b Friday 28-Jun-85 22:54 <sup>E 7</sup> Page 11-1  
 Reassign <Line-id> for service access

|     |        |        |         |        |       |                          |   |
|-----|--------|--------|---------|--------|-------|--------------------------|---|
| 235 | 000262 | 012711 | 140000  |        | MOV   | #ZS.BSY,(R1)             | ; Mark channel temporarily busy         |
| 236 | 000266 | 016702 | 000010G |        | MOV   | \$PBLK+T\$SLN,R2         | ; Get SLN and logical tributary         |
| 237 | 000272 | 016764 | 000016G | 000024 | MOV   | \$PBLK+T\$ASN,C.BUF2(R4) |   |
| 238 | 000300 | 001457 |         |        | BEQ   | CTLAST                   | ; If EQ, no current assignment          |
| 239 |        |        |         |        |       |                          |   |
| 240 | 000302 | 112764 | 000000  | 000004 | MOVB  | #NX.ASS,C.NSP(R4)        |   |
| 241 | 000310 | 016764 | 000016G | 000006 | MOV   | \$PBLK+T\$ASN,C.LIN(R4)  |   |
| 242 | 000316 | 012764 | 002006  | 000010 | MOV   | #FC.CTL!FS.STP,C.FNC(R4) |   |
| 243 | 000324 | 012764 | 000001  | 000012 | MOV   | #STOP,C.STS(R4)          | ; Format CCB to stop current assignment |
| 244 | 000332 |        |         |        | CALLR | QCCBL                    | ; Queue CCB to current LLC              |

|                  |                 |                  |                  |                 |
|------------------|-----------------|------------------|------------------|-----------------|
| ASSIGN= ***** GX | CS.DCR= 000400  | C.LNK 000000     | FS.MNT= 004000   | LF\$CHN= 004000 |
| ASSCHK= 000000   | CS.DEF= 000004  | C.MOD 000011     | FS.MSN= 014000   | LF\$ENB= 000400 |
| ASSCPS= 000000   | CS.DEV= 000002  | C.NSP 000004     | FS.REA= 001000   | LF\$MOP= 001000 |
| ASSPRI= 000000   | CS.DIS= 000040  | C.PRO 000042     | FS.RET= 000000   | LF\$RSR= 100000 |
| ASSTRP= 000000   | CS.ENA= 000001  | C.RSV 000002     | FS.REZ= 003000   | LF.ACT= 100000  |
| CB.CCB= 000002   | CS.ENB= 000020  | C.STA 000007     | FS.RLB= 002000   | LF.BRO= 000400  |
| CB.DDM= 000040   | CS.ERR= 100000  | C.STS 000012     | FS.RNG= 011000   | LF.BWT= 000007  |
| CB.DLC= 000020   | CS.FTL= 001000  | C.URM 177776     | FS.RST= 000000   | LF.ENA= 002000  |
| CB.RDB= 000004   | CS.HCR= 000001  | C.XACP 000004    | FS.RTN= 001000   | LF.LPB= 001000  |
| CB.SDB= 000010   | CS.HFE= 002000  | C.XID 000035     | FS.SET= 005000   | LF.MDC= 000100  |
| CB.SLI= 000100   | CS.LST= 040000  | C.XLEN 000044    | FS.SFC= 005000   | LF.MFL= 004000  |
| CB.XLB= 000001   | CS.MTL= 004000  | C.XPLI 000040    | FS.SFR= 006000   | LF.MTP= 000020  |
| CCBGT= ***** GX  | CS.RNG= 000010  | C.XPT 000034     | FS.SFS= 004000   | LF.PAC= 000200  |
| CCBRT= ***** GX  | CS.ROV= 000004  | C.XSVC 000042    | FS.SPW= 040000   | LF.RDY= 040000  |
| CCPTBL 000776R   | CS.RSN= 010000  | C.XTC 000037     | FS.STM= 000000   | LF.REA= 010000  |
| CC.LLC= 000200   | CS.SHU= 000001  | C.X25 000036     | FS.STP= 002000   | LF.SER= 000040  |
| CE.ABO= 100362   | CS.SID= 000002  | DEASSN= ***** GX | FS.STR= 001000   | LF.TIM= 000010  |
| CE.DAO= 100346   | CS.STR= 000004  | D\$BUG= 177514   | FS.TRM= 003000   | LF.UNL= 020000  |
| CE.DIS= 100366   | CS.SUC= 000001  | D\$ISK= 000000   | FS.WLB= 001000   | LF.X2P= 000000  |
| CE.ERR= 100370   | CS.TMO= 020000  | D\$LL1= 000001   | FS.XKL= 002000   | LLCRS= ***** GX |
| CE.ILN= 100350   | CS.XUR= 000004  | D\$SYNC= 000000  | FS.XOF= 010000   | LN.CLO= 000000  |
| CE.LTO= 100356   | CTLABO 001142R  | D\$SYNM= 000000  | FS.XON= 007000   | LN.DUM= 000005  |
| CE.MOP= 100372   | CTLASS 000404R  | D.BIAS 000000    | FS.ZER= 002000   | LN.LOA= 000004  |
| CE.NTE= 100361   | CTLAS1 000404R  | D.SIZE 000004    | F\$LLVL= 000001  | LN.LOD= 000003  |
| CE.RTE= 100376   | CTLCCP 000742RG | ENBX = 000002 G  | G\$STPP= 000000  | LN.DAU= 000003  |
| CE.SRC= 100364   | CTLDEA 000532R  | E\$XPR= 000000   | G\$STSS= 000000  | LN.OFF= 000001  |
| CE.STP= 100352   | CTLDE1 001106R  | FC.CCP= 000020   | G\$STTK= 000000  | LN.ON= 000000   |
| CE.TME= 100354   | CTLDON 001160R  | FC.CTL= 000006   | G\$SWRD= 000000  | LN.OOP= 000004  |
| CE.TMO= 100374   | CTLERR 000400R  | FC.KCP= 000016   | IEABO 000344R    | LN.OPE= 000001  |
| CE.UNS= 100344   | CTLST 000646R   | FC.KIL= 000004   | IEALC 000360R    | LN.REF= 000002  |
| CF.CHN= 000001   | CTLOFF 000500R  | FC.MAN= 000024   | IENOD 000374R    | LN.SER= 000002  |
| CF.EOM= 000004   | CTLOF1 000554R  | FC.MLD= 000026   | IERSU 000366R    | LN.STA= 000017  |
| CF.HDR= 000020   | CTLOF2 001116R  | FC.PCT= 000030   | IEWLK 000352R    | LN.SUB= 000360  |
| CF.LB= 100000    | CTLON 000130R   | FC.PWR= 000022   | IE.ABO= ***** GX | LN.TRI= 000006  |
| CF.LIN= 000002   | CTLON1 000452R  | FC.RCE= 000002   | IE.ALC= ***** GX | LR\$CTL= 000003 |
| CF.SOM= 000010   | CTLON2 000604R  | FC.RCP= 000014   | IE.MOD= ***** GX | LR\$DEA= 000200 |
| CF.SYN= 000040   | CTLREA 000232R  | FC.TIM= 000010   | IE.NOD= ***** GX | LR\$DIS= 000020 |
| CF.TRN= 000100   | CTLRE1 001010R  | FC.XCP= 000012   | IE.ONP= ***** GX | LR\$DAT= 000010 |
| CLACT= ***** GX  | CTLRE2 001152R  | FC.XME= 000000   | IE.RSU= ***** GX | LS\$DAT= 000004 |
| CM.CIR= 000002   | CTLSER 000172R  | FINDC = ***** GX | IE.WLK= ***** GX | LS\$ERR= 000200 |
| CM.FMT= 100000   | CTLSST 000654R  | FINDL = ***** GX | INIBF = ***** GX | LS\$FRE= 000000 |
| CM.HRD= 000002   | C\$CKP= 000000  | FNCTBL 000022R   | ISALT 000336R    | LS\$IDL= 000001 |
| CM.LIN= 000000   | C\$ORE= 000400  | FNDPR = ***** GX | ISUC = ***** GX  | LS\$MSK= 000037 |
| CM.LOD= 000001   | C\$RSH= 177564  | FNDPR = ***** GX | IS.SET= ***** GX | LS\$RST= 000020 |
| CM.XLO= 000004   | C.ADD 000034    | FS.AST= 000000   | IS.SUC= ***** GX | LS\$STP= 000010 |
| CP.DCF= 000040   | C.BID 000003    | FS.CIB= 002000   | IS\$RAR= 000000  | LS\$STR= 000002 |
| CP.HDL= 000007   | C.BUF 000014    | FS.CRA= 001000   | IS\$RDN= 000000  | LS\$CTL 000012  |
| CP.PS= 177400    | C.BUF1 000014   | FS.DIS= 013000   | I.FCN = ***** GX | LS\$ERR 000010  |
| CP.PSI= 000200   | C.BUF2 000024   | FS.DVC= 001000   | I.PRM = ***** GX | LS\$FLG 000001  |
| CP.XCF= 000100   | C.CNT 000020    | FS.ENO= 012000   | K\$CNI= 177546   | LS\$FNC 000014  |
| CP.2FR= 000030   | C.CNT1 000020   | FS.EXI= 001000   | K\$CSR= 177546   | LS\$LEN 000026  |
| CS.ABO= 000100   | C.CNT2 000030   | FS.GET= 006000   | K\$LDC= 000000   | LS\$LIN 000006  |
| CS.BRO= 000002   | C.FLG 000022    | FS.HLT= 000000   | K\$TPS= 000074   | LS\$LST 000000  |
| CS.BUF= 000200   | C.FLG1 000022   | FS.INI= 000000   | LDIS = 000000 G  | LS\$LUN 000024  |
| CS.CES= 000002   | C.FLG2 000032   | FS.KIL= 000000   | LD\$LP = 000000  | LS\$SRC 000020  |
| CS.CHN= 000010   | C.FNC 000010    | FS.LCL= 100300   | LENB = 000001 G  | LS\$RSR 000002  |
| CS.CMP= 000200   | C.LIN 000006    | FS.LTM= 001000   | LF\$BRO= 002000  | LS\$STT 000007  |

E 9  
C\_XDAT - DLX data bases MACRO V05.03b Friday 28-Jun-85 22:54 Page 8  
Physical link access block storage

```
132                                     .SBTTL Physical link access block storage
133                                     ;+
134                                     ; This area is automatically extended by NTL to include space for all
135                                     ; channels present in the DLX PDV.
136                                     ; -
137
138 000000                                     .PSECT $$$DLX
139
140 000000 PLAST::                               ; Start of physical link access blocks
141
142      000001                                     .END
```

|         |          |          |          |         |        |          |        |         |            |
|---------|----------|----------|----------|---------|--------|----------|--------|---------|------------|
| ABORG   | 000046RG | CS.DIS=  | 000040   | D.BIAS  | 000000 | I\$SRAR= | 000000 | L\$LEN  | 000026     |
| A\$CHK= | 000000   | CS.ENA=  | 000001   | D.SIZE  | 000004 | I\$SRDN= | 000000 | L\$LIN  | 000006     |
| A\$CP=  | 000000   | CS.ENB=  | 000020   | E\$XPR= | 000000 | K\$CNT=  | 177546 | L\$LIST | 000000     |
| A\$PRI= | 000000   | CS.ERR=  | 100000   | FC.CCP= | 000020 | K\$CSR=  | 177546 | L\$LUN  | 000024     |
| A\$TRP= | 000000   | CS.FTL=  | 001000   | FC.CTL= | 000006 | K\$LD=   | 000000 | L\$RCV  | 000020     |
| CB.CCB= | 000002   | CS.HCR=  | 000001   | FC.KCP= | 000016 | K\$TPS=  | 000074 | L\$RSR  | 000002     |
| CB.DDM= | 000040   | CS.HFE=  | 002000   | FC.KIL= | 000004 | LD\$LP=  | 000000 | L\$STT  | 000007     |
| CB.DLC= | 000020   | CS.LST=  | 040000   | FC.MAN= | 000024 | LF\$BRO= | 002000 | L\$TCB  | 000022     |
| CB.RDB= | 000004   | CS.MTL=  | 004000   | FC.MLD= | 000026 | LF\$CHN= | 004000 | L\$TIM  | 000004     |
| CB.SDB= | 000010   | CS.RNG=  | 000010   | FC.PCT= | 000030 | LF\$ENB= | 000400 | L\$TIMI | 000005     |
| CB.SLI= | 000100   | CS.ROV=  | 000004   | FC.PWR= | 000022 | LF\$MOP= | 001000 | L\$TIP  | 000003     |
| CB.XLB= | 000001   | CS.RSN=  | 010000   | FC.RCE= | 000002 | LF\$RSR= | 100000 | L\$XMT  | 000016     |
| CCBGT=  | ***** GX | CS.SHU=  | 000001   | FC.RCP= | 000014 | LF.ACT=  | 100000 | L\$ASG= | 000000     |
| CC.LLC= | 000200   | CS.SID=  | 000002   | FC.TIM= | 000010 | LF.BRO=  | 000400 | L\$DRV= | 000000     |
| CE.ABO= | 100362   | CS.STR=  | 000004   | FC.XCP= | 000012 | LF.BWT=  | 000007 | L\$P11= | 000001     |
| CE.DAO= | 100346   | CS.SUC=  | 000001   | FC.XME= | 000000 | LF.ENA=  | 002000 | L\$11R= | 000000     |
| CE.DIS= | 100366   | CS.TMO=  | 020000   | FS.AST= | 000000 | LF.LPB=  | 001000 | L\$COST | 000015     |
| CE.ERR= | 100370   | CS.XUR=  | 000004   | FS.CIB= | 002000 | LF.MDC=  | 000100 | L\$CTL  | 000012     |
| CE.ILN= | 100350   | C\$CKP=  | 000000   | FS.CRA= | 001000 | LF.MFL=  | 004000 | L\$CVA  | 177776     |
| CE.LTO= | 100356   | C\$ORE=  | 000400   | FS.DIS= | 013000 | LF.MTP=  | 000020 | L\$DDM  | 000002     |
| CE.MOP= | 100372   | C\$RSH=  | 177564   | FS.DVC= | 001000 | LF.PAC=  | 000200 | L\$DDS  | 000004     |
| CE.NTE= | 100361   | C.ADD    | 000034   | FS.ENB= | 012000 | LF.RDY=  | 040000 | L\$DLC  | 000003     |
| CE.RTE= | 100376   | C.BID    | 000003   | FS.EXI= | 001000 | LF.REA=  | 010000 | L\$DLM  | 000006     |
| CE.SRC= | 100364   | C.BUF    | 000014   | FS.GEI= | 006000 | LF.SER=  | 000040 | L\$DLS  | 000010     |
| CE.STP= | 100352   | C.BUF1   | 000014   | FS.HLT= | 000000 | LF.TJM=  | 000010 | L\$FLG  | 000000     |
| CE.TME= | 100354   | C.BUF2   | 000024   | FS.INI= | 000000 | LF.UNL=  | 020000 | L\$KREA | 000016     |
| CE.TMO= | 100374   | C.CNT    | 000020   | FS.KIL= | 000000 | LF.X2P=  | 000000 | L\$LEN  | = 000022   |
| CE.UNS= | 100344   | C.CNT1   | 000020   | FS.LCL= | 100000 | LN.CLO=  | 000000 | L\$MPF  | 000022     |
| CF.CHN= | 000001   | C.CNT2   | 000030   | FS.LTM= | 001000 | LN.DUM=  | 000005 | L\$NMST | 000020     |
| CF.EOM= | 000004   | C.FLG    | 000022   | FS.MNT= | 004000 | LN.LOA=  | 000004 | L\$NSTA | 000014     |
| CF.HDR= | 000020   | C.FLG1   | 000022   | FS.MSN= | 014000 | LN.LOO=  | 000003 | L\$OWNR | 000021     |
| CF.LB   | = 100000 | C.FLG2   | 000032   | FS.KCA= | 001000 | LN.OAU=  | 000003 | L\$UNT  | 000013     |
| CF.LIN= | 000002   | C.FNC    | 000010   | FS.RE=  | 000000 | LN.OFF=  | 000001 | M\$CRB= | 000124     |
| CF.SOM= | 000010   | C.LIN    | 000006   | FS.REZ= | 003000 | LN.ON=   | 000000 | M\$CRX= | 000000     |
| CF.SYN= | 000040   | C.LNK    | 000000   | FS.RLB= | 002000 | LN.OOP=  | 000004 | M\$FCS= | 000000     |
| CF.TRN= | 000100   | C.MOD    | 000011   | FS.RNG= | 011000 | LN.OPE=  | 000001 | M\$MGE= | 000000     |
| CM.CIR= | 000002   | C.NSP    | 000004   | FS.RTN= | 000000 | LN.REF=  | 000002 | M\$NET= | 000000     |
| CM.FMT= | 100006   | C.PRO    | 000042   | FS.SET= | 001000 | LN.SER=  | 000002 | M\$OVR= | 000000     |
| CM.HRD= | 000002   | C.RSV    | 000002   | FS.SFC= | 005000 | LN.STA=  | 000017 | N\$ACC= | 000001     |
| CM.LIN= | 000000   | C.STA    | 000007   | FS.SFR= | 006000 | LN.SUB=  | 000360 | N\$BUF= | 000001     |
| CM.LOO= | 000001   | C.STS    | 000012   | FS.SFS= | 004000 | LN.TRI=  | 000006 | N\$LDV= | 000001     |
| CM.VLO= | 000004   | C.URM    | 177776   | FS.SPW= | 040000 | LR\$CTL= | 000003 | N\$MCP= | 000001     |
| CP.DCF= | 000040   | C.XACP   | 000004   | FS.STM= | 000000 | LR\$DIS= | 000020 | N\$MLL= | 000001     |
| CP.HDL= | 000007   | C.XID    | 000035   | FS.STP= | 002000 | LR\$DIP= | 000010 | N\$MOV= | 000010     |
| CP.PS   | = 177400 | C.XLEN   | 000044   | FS.STR= | 001000 | LR\$STP= | 000010 | N\$NCT= | 000001     |
| CP.PSI= | 000200   | C.XPLI   | 000040   | FS.TRM= | 003000 | L\$SDAT= | 000004 | N\$PEM= | 000001     |
| CP.XCF= | 000100   | C.XPT    | 000034   | FS.WLB= | 001000 | L\$SERR= | 000200 | P\$P45= | 000000     |
| CP.2FR= | 000030   | C.XSVC   | 000042   | FS.XKL= | 002000 | L\$SFRE= | 000000 | P\$WRD= | 000000     |
| CS.ABO= | 000100   | C.XTC    | 000037   | FS.XOF= | 010000 | L\$SIDL= | 000001 | Q\$OPT= | 000010     |
| CS.BRO= | 000002   | C.X25    | 000036   | FS.XON= | 007000 | L\$MSK=  | 000037 | RCVAB   | = ***** GX |
| CS.BUF= | 000200   | DISRQ    | 000066RG | FS.ZER= | 002000 | L\$RST=  | 000020 | R\$DER= | 000000     |
| CS.CES= | 000002   | DOFNC    | 000000RG | F\$LVL= | 000001 | L\$STP=  | 000010 | R\$K11= | 000001     |
| CS.CHN= | 000010   | D\$BUG=  | 177514   | G\$TPP= | 000000 | L\$STR=  | 000002 | R\$SND= | 000000     |
| CS.CMP= | 000200   | D\$ISK=  | 000000   | G\$TSS= | 000000 | L\$CTL   | 000012 | R\$11M= | 000000     |
| CS.DCR= | 000400   | D\$SL1=  | 000001   | G\$TTK= | 000000 | L\$ERR   | 000010 | SF.ACT= | 000200     |
| CS.DEF= | 000004   | D\$SYNC= | 000000   | G\$WRD= | 000000 | L\$FLG   | 000001 | SF.ENA= | 000100     |
| CS.DEV= | 000002   | D\$SYNM= | 000000   |         |        | L\$FNC   | 000014 | SF.LPB= | 000004     |

```

; Flags word bit definitions for X.25 circuits and lines (L$FLX)
;
; Circuits:
000001      L$CHN = 1      ; Set PVC channel
000002      L$CMB = 2      ; Set maximum data
000004      L$CUS = 4      ; Set circuit usage
000010      L$DTE = 10     ; Set DTE
000020      L$MXR = 20     ; Set maximum recalls
000040      L$MXW = 40     ; Set maximum window
000100      L$NUM = 100    ; Set DLM number
000200      L$RET = 200    ; Set recall timer
001000      L$BLK = 1000   ; Set blocking
100000      L$NXC = 100000 ; Circuit is a new X.25 circuit

; Lines:
000400      L$HBT = 400    ; Set holdback timer
000002      L$LMB = L$CMB  ; Set max block
000020      L$MRT = L$MXR  ; Set max retransmits
000040      L$MWN = L$MXW  ; Set max window
000200      L$NTI = L$RET  ; Set retransmit timer
010000      L$PRO = 10000  ; Set line protocol

; Common:
020000      L$LCT = 20000  ; Set counter timer

; Groupings:
020053      L$PVC = L$CHN!L$CMB!L$DTE!L$MXW!L$LCT
000362      L$SVC = L$CMB!L$MXR!L$MXW!L$NUM!L$RET
001004      L$DLM = L$BLK!L$CUS
  
```

DLXSUB - DLX subroutines  
Table of contents

MACRO V05.03b Friday 28-Jun-85 22:56 E 12

|     |      |   |
|-----|------|---|
| 5-  | 42   | Macro definitions   |
| 6-  | 53   | Assign line to an LLC process                                 |
| 7-  | 115  | Validate optional characteristics buffer and relocate address |
| 8-  | 169  | Convert ASCII to RAD50  |
| 9-  | 227  | Convert decimal to binary                                     |
| 10- | 271  | Validate characteristics buffer                               |
| 11- | 309  | Compute address in the COMM/EXEC databases                    |
| 12- | 347  | Deassign line from an LLC                                     |
| 13- | 401  | Deassign physical link  |
| 14- | 417  | Allocate and fill a chain of large data buffers               |
| 15- | 492  | Find a free broadcast port                                    |
| 16- | 535  | Validate privileged user and search for <Line-id>             |
| 17- | 641  | Find free channel in process mapping table                    |
| 18- | 701  | Scan databases for <Line-id>                                  |
| 19- | 794  | Find <Line-id> in the COMM/EXEC database                      |
| 20- | 885  | Find an LLC process PDV index                                 |
| 21- | 917  | Get SLN and logical tributary #                               |
| 22- | 942  | Compute address within process channel mapping table          |
| 23- | 971  | Check if process is loaded                                    |
| 24- | 996  | Release a chain of receive data buffers                       |
| 25- | 1011 | Compute address of physical link access block                 |
| 26- | 1041 | Compute address of network management state byte              |
| 27- | 1075 | Compute address of the system line table                      |



DLXSUB - DLX subroutines      MACRO V05.03b Friday 28-Jun-85 22:56      Page 14-1  
 Allocate and fill a chain of large data buffers

```

474 001076 060067 000000G      ADD    R0,$VA      ; Update source virtual address
475 001102 160067 000000G      SUB    R0,$BYTES   ; and remaining byte count
476 001106 001323              BNE    10$      ; If NE, more to go
477                               ;
478 001110 012604              MOV    (SP)+,R4    ; Get address chain
479 001112              RESRG   <R5>      ; Restore physical link block address
480 001114 042715 100000      BIC    #LF$RSR,(R5) ; Show no allocation failure
481 001120 000241              CLC          ; Indicate success
482 001122              RETURN
483                               ;
484 001124 012604      100$: MOV    (SP)+,R4    ; Get address of next CCB in chain
485 001126 001402      BEQ    110$      ; If EQ, no more
486 001130              CALL    RLSCHN      ; Release the chain
487                               ;
488 001134      110$: RESRG   <R5>      ; Restore physical link block address
489 001136 000261      SEC          ; Indicate allocation failure
490 001140              RETURN
  
```

DLXSUB - DLX subroutines  
Check if process is loaded

MACRO V05.03b Friday 28-Jun-85 22:56 Page 23

E 14

```
971 .SBTTL Check if process is loaded
972 ;+
973 ;**--PRCLD-Check if process is loaded
974 ;
975 ; Test if the specified process is loaded.
976 ;
977 ; Inputs:
978 ; R5 = PDV and Channel
979 ;
980 ; Outputs:
981 ; R0 = Error code (IE.MOD) if process is not loaded
982 ; 'C' Clear - Process is loaded
983 ; 'C' Set - Process is not loaded
984 ;
985 ; Registers modified:
986 ; R0, R1, R2
987 ;
988 002416 016700 000016G PRCLD:: MOV $PBLK+T$ASN,R0 ; Get the PDV and Channel assignment
989 002422 CALL MAPAD ; Compute the PDV address
990 002426 005762 000012 TST Z.PCB(R2) ; Is the process loaded?
991 002432 001003 BNE 10$ ; If NE. yes
992 002434 012700 000000C MOV #IE.MOD&377,R0 ; Set up error code
993 002440 000261 SEC ; Indicate process not loaded
994 002442 10$: RETURN
```

[illegible]

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LL          SSSSSS    TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LLLLLLLLLL SSSSSSSS  TT
LLLLLLLLLL SSSSSSSS  TT

```

```

381
382
383
384
385
386
387
388
389
390
391
392
393
394
395 000636
396 000640 116302 000025
397 000644 042702 177600
398 000650 067702 177156
399 000654 011202
400 000656 005001
401 000660 156301 000024
402 000664 006301
403 000666 060102
404 000670 016263 000020 000024
405 000676
406 000700

.SBTTL MAP PDV & CHANNEL TO SLN & STATION NUMBER
+
**MAPCHN-MAP PDV AND CHANNEL PAIR TO SLN AND STATION NUMBER PAIR
LLC PROCESSES OPERATE IN TERMS OF CHANNEL NUMBERS. SINCE LINES MAY
BE DYNAMICALLY SWITCHED BETWEEN LLC PROCESSES, WE MUST CONVERT THIS
PAIR INTO A SLN AND STATION NUMBER PAIR.
-
INPUTS:
R3 - ADDRESS OF EVENT RECORD
REGISTERS MODIFIED:
R2
MAPCHN: SAVRG <R1> : GET A FREE REGISTER
        MOVB E.LIN+1(R3),R2 : GET PROCESS' PDV INDEX
        BIC #*C<177>,R2 : AND REMOVE UNWANTED BITS
        ADD @PDVTA,R2 : POINT INTO PDV INDEX TABLE
        MOV (R2),R2 : AND GET ADDRESS OF PDV
        CLR R1 : GET CHANNEL NUMBER
        BISB E.LIN(R3),R1 :
        ASL R1 : FORM WORD OFFSET
        ADD R1,R2 : INTO PDV MAPPING TABLE
        MOV Z.MAP(R2),E.LIN(R3)
        RESRG <R1>
        RETURN

```

```

111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129 000056
130 000064
131 000070
132 000100
133
134 000104 004167 000000G
135 000110 016
136 000111 050
137 000112 103424
138
139 000114 116722 000000G
140 000120 112722 000002
141 000124 017701 000000G
142 000130 116122 000036
143 000134 116122 000037
144 000140
145 000144
146 000154 112713 000004
147 000160 000241
148 000162
149
150 000164

.SBTTL Accept incoming connect request
;+
**--SNDACC-Accept incoming connect request
; This routine will allocate the resources to transmit a connect
; confirm message to accept the logical link.
;
; Inputs:
; R3 = Virtual address of LLT
; R5 = Address of database descriptor
;
; Outputs:
; 'C' Clear - Connect confirm message sent
; 'C' Set - Resource allocation failure
;
; Registers modified:
; R0, R1, R2, R4
SNDACC::MAPLLT ; Map to the LLT
CALL SETREQ ; Set up connect request services
CALLLE RTRSES ; Start round trip timer
CALL SAVOPT ; Save optional data in the LLT
;
JSR R1,GETSDB ; Allocate a small message buffer
; Connect confirm message '
; BYTE NT$CC
; BYTE MF,CTL!MC.CC
BCS 100$ ; If CS, allocation failure
;
MOVB $SRVCS,(R2)+ ; Fill in requested services
MOVB #CV$40,(R2)+ ; Fill in info field - NSP version 4.0
MOV @DECP,R1 ; Point to DEC home block
MOVB D$SEG(R1),(R2)+ ; Fill in segment size
MOVB D$SEG+1(R1),(R2)+ ;
;
CALL ADDOPT ; Add optional data
CALLLE SNSESD ; Transmit the data
MOVB #ST$CC,(R3) ; Set state to connect confirm sent
CLC ; Indicate success
RETURN
100$: CALLR REMLNK ; Remove link data structures and return

```

```

581                                     .SBTTL Complete interrupt message processing
582                                     ;+
583                                     **-CMPINT-Complete processing of an interrupt message
584                                     ;
585                                     The interrupt message data has been copied into the user's buffer.
586                                     release the resources associated with the interrupt message and
587                                     try to send a new link service message for the interrupt
588                                     subchannel.
589                                     ;
590                                     -
591                                     Inputs:
592                                     R4 = Address of CCB
593                                     C.STS = Physical address of LLT
594                                     R5 = Address of database descriptor
595                                     ;
596                                     Registers modified:
597                                     R0, R1, R2, R3, R4
598 001414 112764 000004 000011 CMPINT::MOVB #NT$INT,C.MOD(R4)
599 001422 016403 000012      MOV C.STS(R4),R3      ; Get physical address of LLT
600 001426      CALLE FLSRES      ; Flush the resources
601
602 001436      CALLE ACCLLT      ; Gain access to LLT
603 001446 005263 000044      INC L.LSFI(R3)      ; Update 1/LS flow control count
604 001452      CALLE CHKACK      ; Try to send the link service message
605
606 001462      RETURN
607

```

|     |     |                                     |
|-----|-----|-------------------------------------|
| 5-  | 42  | Macro definitions                   |
| 6-  | 53  | Define local macros                 |
| 7-  | 60  | Process dispatch table              |
| 8-  | 77  | Control enable processing           |
| 9-  | 95  | Start or line assignment            |
| 10- | 142 | Line shutdown                       |
| 11- | 182 | Timeout processing                  |
| 12- | 232 | Transmit complete processing        |
| 13- | 268 | Receive complete processing         |
| 14- | 335 | Control complete processing         |
| 15- | 387 | Asynchronous error processing       |
| 16- | 409 | Enable complete processing          |
| 17- | 427 | Start complete processing           |
| 18- | 461 | Characteristics complete processing |
| 19- | 481 | Stop complete processing            |

```

387                                     .SBTTL Asynchronous error processing
388
389                                     ;+
390                                     **--ASYNCH-Asynchronous error processing
391                                     ;
392                                     This routine is called when an asynchronous error is reported by the
393                                     DLC layer.
394                                     ;
395                                     -
396                                     Inputs:
397                                     R1 = Completion status
398                                     R2 = Error offset for characteristics functions
399                                     R5 = Address of physical link access block
400
401 ASYNCH: CMPB    R1,#CE.DIS           ; Is this a disconnect error?
402          BEQ     10$                 ; If EQ, yes
403          TST     L$TCB(R5)           ; Is there an active user for this link?
404          BEQ     20$                 ; If EQ, no
405          CALLR   STPRQ               ; Force error on channel
406
407          10$: CALLR   DISRQ           ; Force a disable to be issued
408
409          20$: RETURN

```



DLXCEX      CREATED BY    MACRO    ON 28-JUN-85 AT 22:53      PAGE 6      F 5

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES   |
|------------|--|
| CALL       | 8-88      9-120      9-121      10-161      10-162      10-179      11-205      11-214      12-248      12-260     |
|            | 12-264      13-283      13-311      13-320      13-322      13-328      14-357      14-366      16-423      17-454 |
|            | 19-493      19-531      19-534      19-550   |
| CALLR      | 8-89      9-140      10-180      12-266      13-326      14-367      14-369      15-403      15-405      17-457    |
|            | 18-477      19-542   |
| CCBDF\$    | #5-45      5-47  |
| CHRD\$     | #5-51      5-51  |
| DLXDF\$    | #5-45      5-51  |
| MAP        | #5-44      13-318  |
| PDVDF\$    | #5-45      5-48  |
| PLADF\$    | #5-45      5-49  |
| RESRG      | #5-44      10-164      13-316  |
| RETURN     | 11-221      13-333      15-407      16-425      17-459      18-479      19-553                                     |
| SAVRG      | #5-44      10-160      13-292  |
| SLTDF\$    | #5-45      5-50  |
| \$DOPKT    | #6-55      17-448      18-475  |

DLXCHR      CREATED BY    MACRO    ON 28-JUN-85 AT 22:53      PAGE 1      F 6

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES   |
|---------|------------|--|
| CC.ADR  | = 000100   | 5-82      9-244  |
| CS.IGN  | = 000002   | 5-80      9-242  |
| CS.SUC  | = 000001   | 6-137      7-171      8-217      10-287      11-311  |
| C.ADD   | 000034     | 6-133      6-134      6-135  |
| C.BUF   | 000014     | 8-200  |
| C.CHRL  | 000010     | 5-92      *6-133      *6-134      *6-135      *7-169      8-210      9-253      10-283      10-284 |
|         |            | 10-285      11-309   |
| C.DATI  | 000002     | 5-91      6-127      7-163      8-197      9-252      10-280      11-306                           |
| C.DATO  | 000004     | *5-87      *6-136      *7-170      *8-216  |
| C.NSP   | 000004     | 8-201  |
| C.PRO   | 000042     | 7-169  |
| C.STAT  | 000006     | *5-90      *9-251  |
| FL\$ADR | = ***** GX | 10-286   |
| FL\$PRO | = ***** GX | 11-310   |
| I.PRM   | = ***** GX | 5-78      5-93      8-208      9-240      9-254  |
| KISAR6  | = ***** GX | *8-200      *8-208   |
| LF\$BRO | = 002000   | 6-130      7-166   |
| N\$SVCT | = *****    | 8-200      8-208   |
| RCMAX   | = 000006   | 5-85      #5-103   |
| RCVADR  | 000072 R   | 5-100      #6-127  |
| RCVCHR  | 000000 RG  | #5-78  |
| RCVDAD  | 000206 R   | 5-102      #8-197  |
| RCVPRO  | 000146 R   | 5-101      #7-163  |
| RCVTBL  | 000064 R   | 5-88      #5-100      5-103  |
| XMMAX   | = 000004   | 9-247      #9-263  |
| XMTADR  | 000400 R   | 9-261      #10-280   |
| XMTCHR  | 000314 RG  | #9-240   |
| XMTPRO  | 000446 R   | 9-262      #11-306   |
| XMTTBL  | 000374 R   | 9-249      #9-261      9-263      *10-285  |
| \$ADR   | = ***** GX | *10-283      *10-284      *10-285  |
| \$FLAGS | = ***** GX | *10-286  |
| \$PRO   | = ***** GX | *11-309  |

```

246                                     .SBTTL Common error routines
247                                     ;+
248                                     ;*-IEXXX-Common error returns
249                                     ;*-ISXXX-Alternate success returns
250                                     ;
251                                     Complete the I/O requests with common error codes.
252                                     ;
253                                     Inputs:
254                                     R3 = Address of I/O packet
255                                     ;
256                                     Registers modified:
257                                     R0, R1, R2, R3, R4, R5
258                                     ;
259                                     .ENABL LSB
260
261 000336 012700 000000C ISALT: MOV #IS.SET&377,R0 ; Set alternate success status
262 000342 000416          BR 10$
263
264 000344 012700 000000C IEABO: MOV #iE.ABO&377,R0 ; Set line in wrong state
265 000350 000413          BR 10$
266
267 000352 012700 000000C IEWLK: MOV #iE.WLK&377,R0 ; Set access denied
268 000356 000410          BR 10$
269
270 000360 012700 000000C IEALC: MOV #iE.ALC&377,R0 ; Set allocation error status
271 000364 000405          BR 10$
272
273 000366 012700 000000C IERSU: MOV #iE.RSU&377,R0 ; Set shared resource in use
274 000372 000402          BR 10$
275
276 000374 012700 000000C IENOD: MOV #iE.NOD&377,R0 ; Set secondary resource allocation failure
277
278 000400 10$:
279 000400 CTLERR: CALLR RQALT ; Post completion on I/O packet
280
281                                     .DSABL LSB

```

DLXCTL - DLx control Q10 functi MACRO V05.03b Friday 28-Jun-85 22:54 Page 23-2  
Symbol table

|                  |                  |                  |                   |                   |
|------------------|------------------|------------------|-------------------|-------------------|
| L\$TCB 000022    | M\$SCRX= 000000  | R\$SDER= 000000  | T\$TBF = ***** GX | ZF.KMX= 000020    |
| L\$TIM 000004    | M\$SFCS= 000000  | R\$SK11= 200001  | T\$ASN 000016     | ZF.LLC= 000004    |
| L\$IMI 000005    | M\$SMGE= 000000  | R\$SSND= 000000  | T\$CTL 000002     | ZF.LMC= 000100    |
| L\$IP 000003     | M\$NET= 000000   | R\$S11M= 000000  | T\$DDM 000014     | ZF.MAN= 020000    |
| L\$XMT 000016    | M\$SOVR= 000000  | SFMAX = 000022   | T\$DLC 000015     | ZF.MFL= 000010    |
| L\$SASG= 000000  | NXCTL 000044RG   | SF.ACT= 000200   | T\$FLG 000006     | ZF.MTM= 000400    |
| L\$SDRV= 000000  | NX.ASS= 000000 G | SF.ENA= 000100   | T\$KRB 000020     | ZF.MUX= 000040    |
| L\$SP11= 000001  | NX.DEA= 000006 G | SF.LPB= 000004   | T\$LEN 000024     | ZF.PSE= 002000    |
| L\$S11R= 000000  | NX.DON= 000010 G | SF.MFL= 000040   | T\$LIN 000003     | ZF.SLI= 010000    |
| L.COST 000015    | NX.OFF= 000004 G | SF.PAC= 000020   | T\$NAM 000000     | ZF.TIM= 000200    |
| L.CTL 000012     | NX.REA= 000002 G | SF.REA= 000010   | T\$NMST 000022    | ZF.X3P= 000000    |
| L.CVA 177776     | N\$SACC= 000001  | SF.SER= 000001   | T\$NSTA 000005    | ZS.ASN= 100000    |
| L.DDM 000002     | N\$SBUF= 000001  | SF.SVC= 000002   | T\$PDV 000007     | ZS.BSY= 140000    |
| L.DDS 000004     | N\$SLDV= 000001  | SF.UNL= 000040   | T\$SFLG 000004    | Z.AVL 000014      |
| L.DLC 000003     | N\$SMCP= 000001  | SRCTBL 000000R   | T\$SLF 000012     | Z.DAT 000016      |
| L.DLM 000006     | N\$SMLL= 000001  | STACT = ***** GX | T\$SLN 000010     | Z.DSP 000000      |
| L.DLS 000010     | N\$SMOV= 000010  | STNMST= ***** GX | T\$SKMG= 000000   | Z.FLG 000010      |
| L.FLG 000000     | N\$SNCT= 000001  | STOP = 000001 G  | T\$SMIN= 000000   | Z.LEN = 000016    |
| L.KRBA 000016    | N\$SPEM= 000001  | STRT = 000004 G  | V\$SCTR= 001000   | Z.LLN 000006      |
| L.LEN = 000022   | OFF = 000000 G   | S\$SWRG= 000000  | X\$SDBT= 000000   | Z.MAP 000020      |
| L.MPF 000022     | PRCLD = ***** GX | S\$SYSZ= 007600  | ZF.COU= 001000    | Z.NAM 000004      |
| L.NMST 000020    | P\$SP45= 000000  | S.COST 000001    | ZF.DDM= 000001    | Z.PCB 000012      |
| L.NSTA 000014    | P\$SWRD= 000000  | S.FLG 000000     | ZF.DIA= 004000    | Z.SCH 000007      |
| L.OWNR 000021    | QCCBL 000710RG   | S.LEN 000004     | ZF.DLC= 000002    | \$DLXPD= ***** GX |
| L.UNT 000013     | Q\$SOPT= 000010  | S.NMST 000002    | ZF.DVP= 100000    | \$PBLK = ***** GX |
| MAPAD = ***** GX | ROALT = ***** GX | S.OWNR 000003    | ZF.INI= 040000    | .\$\$\$\$= 00C034 |
| M\$SCRB= 000124  | RQDON = ***** GX |                  |                   |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
001176 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 45  
Work file writes: 53  
Size of work file: 18719 Words ( 74 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:25.13  
SY:DLXCTL.V2,[131,134]DLXCTL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]DLXCTL

DLXDAT - DLX data bases MACRO V05.03b Friday 28-Jun-85 22:54 Page 8-1  
Symbol table

|                   |                 |                |                   |                    |
|-------------------|-----------------|----------------|-------------------|--------------------|
| ACHCK 000066RG    | K\$CSR= 177546  | L\$RSR 000002  | RDBSZ 000134RG    | \$BIGBF 000024RG   |
| A\$CHCK= 000000   | K\$SLDC= 000000 | L\$STT 000007  | RELOC 000100RG    | \$BIGSZ 000030RG   |
| A\$CPS= 000000    | K\$STPS= 000074 | L\$TCB 000022  | R\$DEFER= 000000  | \$BLXIO= ***** GX  |
| A\$SPRI= 000000   | LDBGIT 000114RG | L\$TIM 000004  | R\$K11= 000001    | \$BUFFG 000032RG   |
| A\$TRP= 000000    | LD\$LP = 000000 | L\$TIM1 000005 | R\$SND= 000000    | \$BYTES 000056RG   |
| BLXIO 000070RG    | LF\$BRO= 002000 | L\$TIP 000003  | R\$TIM= 000000    | \$CCBGT= ***** GX  |
| CCBGT 000104RG    | LF\$CHN= 004000 | L\$XMT 000016  | SLTMA 000136RG    | \$CCBRT= ***** GX  |
| CCBRT 000106RG    | LF\$ENB= 000400 | L\$ASG= 000000 | SLTNM 000140RG    | \$CEMUL= ***** GX  |
| CEMUL 000110RG    | LF\$MOP= 001000 | L\$DRV= 000000 | \$SWRG= 000000    | \$CMPDV= ***** GX  |
| CMPDV 000112RG    | LF\$RSR= 100000 | L\$P11= 000001 | \$SYSZ= 007600    | \$DLXCH 000033RG   |
| C\$CKP= 000000    | LLCRQ 000116RG  | L\$11R= 000000 | T\$TBF 000102RG   | \$DLXPD 000034RG   |
| C\$ORE= 000400    | LLCRS 000120RG  | MAXOV 000124RG | T\$ASN 000016     | \$DLXVL= 000027 G  |
| C\$RSH= 177564    | LLCTA 000122RG  | M\$CRB= 000124 | T\$CTL 000002     | \$DLXVT 000062RG   |
| D\$BUG= 177514    | LR\$CTL= 000003 | M\$CRX= 000000 | T\$DDM 000014     | \$FLAGS 000046RG   |
| D\$ISK= 000000    | LR\$DEA= 000200 | M\$FCS= 000000 | T\$DLC 000015     | \$INIBF= ***** GX  |
| D\$LL1= 000001    | LR\$DIS= 000020 | M\$MGE= 000000 | T\$FLG 000006     | \$IOFIN= ***** GX  |
| D\$SYNC= 000000   | LR\$STP= 000010 | M\$NET= 000000 | T\$KRB 000020     | \$IOS 000060RG     |
| D\$SYNM= 000000   | LS\$DAT= 000004 | M\$OVR= 000000 | T\$LEN 000024     | \$LD\$GT= ***** GX |
| D.BIAS 000000     | LS\$ERR= 000200 | M\$ACC= 000001 | T\$LIN 000003     | \$LLCRQ= ***** GX  |
| D.SIZE 000004     | LS\$FRE= 000000 | N\$BUF= 000001 | T\$NAM 000000     | \$LLCRS= ***** GX  |
| E\$XPR= 000000    | LS\$IDL= 000001 | N\$LDV= 000001 | T\$NMST 000022    | \$LLCTA= ***** GX  |
| FL\$ADR= 000001 G | LS\$MSK= 000037 | N\$MCP= 000001 | T\$NSTA 000005    | \$MAXOV= ***** GX  |
| FL\$PRO= 000002 G | LS\$RST= 000020 | N\$MLL= 000001 | T\$PDV 000007     | \$PBLK 000000RG    |
| FL\$VL= 000001    | LS\$STP= 000010 | N\$MOV= 000010 | T\$SFLG 000004    | \$PDVID= ***** GX  |
| G\$TPP= 000000    | LS\$STR= 000002 | N\$NCT= 000001 | T\$SLF 000012     | \$PDVTA= ***** GX  |
| G\$TSS= 000000    | L\$CTL 000012   | N\$PEM= 000001 | T\$SLN 000010     | \$PRO 000044RG     |
| G\$TTK= 000000    | L\$ERR 000010   | PDVID 000126RG | T\$KMG= 000000    | \$QUEBF= ***** GX  |
| G\$WRD= 000000    | L\$FLG 000001   | PDVTA 000130RG | T\$SMIN= 000000   | \$RDBRT= ***** GX  |
| INIBF 000072RG    | LF\$NC 000014   | PLAST 000000RG | V\$CTR= 001000    | \$RDBSZ= ***** GX  |
| IOFIN 000074RG    | L\$LEN 000026   | P\$P45= 000000 | X\$DBT= 000000    | \$RELOC= ***** GX  |
| I\$RAR= 000000    | L\$LIN 000006   | P\$WRD= 000000 | \$ACCESS 000000RG | \$SLTMA= ***** GX  |
| I\$RDN= 000000    | L\$ST 000000    | QUEBF 000076RG | \$ACHCK= ***** GX | \$SLTNM= ***** GX  |
| KISAR6= ***** GX  | L\$LUN 000024   | Q\$OPT= 000010 | \$ADR 000036RG    | \$TSTBF= ***** GX  |
| KSAR6 000064RG    | L\$RCV 000020   | RDBRT 000132RG | \$BIAS 000052RG   | \$VA 000054RG      |
| K\$CNT= 177546    |                 |                |                   |                    |

. ABS. 000026 000 (RW,I,GBL,ABS,OVR)  
000142 001 (RW,I,LCL,REL,CON)  
\$DLX 000000 002 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 10335 Words ( 41 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:07.35  
SY:DLXDAT.V2,[131,134]DLXDAT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:[1,131,10]DLXDAT

F 10

DLXLIN - DLX Line control routi MACRO V05.03b Friday 28-Jun-85 22:54 Page 8-2

Symbol table

|                  |                |                  |                |                   |
|------------------|----------------|------------------|----------------|-------------------|
| SF.MFL= 000040   | S.LEN 000004   | T\$PDV 000007    | ZF.INI= 040000 | ZS.BSY= 140000    |
| SF.PAC= 000020   | S.NMST 000002  | T\$SFLG 000004   | ZF.KMX= 000020 | Z.AVL 000014      |
| SF.REA= 000010   | S.DWNR 000003  | T\$SLF 000012    | ZF.LLC= 000004 | Z.DAT 000016      |
| SF.SER= 000001   | T\$ASN 000016  | T\$SLN 000010    | ZF.LMC= 000100 | Z.DSP 000000      |
| SF.SVC= 000002   | T\$CTL 000002  | T\$KMG= 000000   | ZF.MAN= 020000 | Z.FLG 000010      |
| SF.UNL= 000040   | T\$DDM 000014  | T\$MIN= 000000   | ZF.MFL= 000010 | Z.LEN = 000016    |
| STOPC = ***** GX | T\$DLC 000015  | V\$CTR= 001000   | ZF.MTM= 000400 | Z.LLN 000006      |
| STPCHN 000156RG  | T\$FLG 000006  | XMTAB = ***** GX | ZF.MUX= 000040 | Z.MAP 000020      |
| STPRQ 000074RG   | T\$KRB 000020  | X\$DBT= 000000   | ZF.PSE= 002000 | Z.NAM 000004      |
| STRCHN 000250RG  | T\$LEN 000024  | ZF.COU= 001000   | ZF.SLI= 010000 | Z.PCB 000012      |
| S\$WRG= 000000   | T\$LIN 000003  | ZF.DDM= 000001   | ZF.TIM= 000200 | Z.SCH 000007      |
| S\$YSZ= 007600   | T\$NAM 000000  | ZF.DIA= 004000   | ZF.X3P= 000000 | \$LLCRQ= ***** GX |
| S.COST 000001    | T\$NMST 000022 | ZF.DLC= 000002   | ZS.ASN= 100000 | .\$\$\$\$= 000034 |
| S.FLG 000000     | T\$NSTA 000005 | ZF.DVP= 100000   |                |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR) -  
 000334 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 51  
 Work file writes: 59  
 Size of work file: 18796 Words ( 74 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:15.53  
 SY:DLXLIN.V2,[131,134]DLXLIN/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]DLXLIN

DLXQ10 - ISSUE I/O TO DLX  
ERROR MESSAGE STRINGS

MACRO V05.03b Saturday 29-Jun-85 12:23 Page 7

F 11

78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89

```
.SBTTL  ERROR MESSAGE STRINGS
;
; ERROR MESSAGE STRINGS
;
; .ENABL  LC
;
; NOTE: THE NEXT THREE LINES MUST REMAIN CONTIGUOUS
;
DLXMSG: .BLKB 1 ; BUFFER FOR DLX STRING LENGTH
DLXTXT: .ASCII /DLX error #/
DLXERR: .BLKB 6 ; BUFFER FOR DLX ERROR CODE
.EVEN
```

104

114

130

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

.TITLE DLXSUB - DLX subroutines  
.IDENT /V05.00/  
.ENABL LC

Copyright (C) 1982, 1983, 1985 by  
Digital Equipment Corporation, Maynard, MASS.

This software is furnished under a license for use only on a  
single computer system and may be copied only with the  
inclusion of the above copyright notice. This software, or  
any other copies thereof, may not be provided or otherwise  
made available to any other person except for use on such  
system and to one who agrees to these license terms. Title  
to and ownership of the software shall at all times remain  
in DEC.

The information in this document is subject to change without  
notice and should not be construed as a commitment by Digital  
Equipment Corporation.

DEC assumes no responsibility for the use or reliability of  
its software on equipment which is not supplied by DEC.

Module description

DLX subroutines

Ident history:

4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0  
  
5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/R SX V1.0



DLXSUB - DLX subroutines  
Find a free broadcast port

MACRO V05.03b Friday 28-Jun-85 22:56 Page 15

F.13

```

492 .SBTTL Find a free broadcast port
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508 001142 116100 000014
509 001146 001414
510
511 001150 116702 000010G
512 001154 006302
513 001156 067702 000000G
514 001162 011202
515 001164 100005
516
517 001166 006302
518 001170 005722
519 001172 001404
520 001174
521 001200 000261
522 001202
523
524 001204 116102 000014
525 001210 160002
526 001212 110267 000011G
527 001216 006302
528 001220 006302
529 001222 060102
530 001224 116267 000022 000004G
531 001232 016267 000024 000022G
532 001240 005067 000016G
533 001244

      Find an unused port on the specified broadcast channel <Line-id>.
Inputs:
      R1 = Address of the system line table
Outputs:
      'C' Clear - Free port found
      'C' Set - No free ports available
Registers modified:
      R0, R2
FNDBR:: MOV B L,NSTA(R1),R0 ; Get # of entries to scan
        BEQ 20$ ; If EQ, none!
        MOV B $PBLK+T$SLN,R2 ; Get the system line #
        ASL R2 ; Compute address of reverse
        ADD @LLCTA,R2 ; mapping table for this
        MOV (R2),R2 ; <Line-id>
        BPL 20$ ; If PL, not set up as multi-point
        ASL R2 ; Form word address
10$: TST (R2)+ ; Is this a free entry?
      BEQ 30$ ; If EQ, yes
      SOB R0,10$ ; Check all entries
20$: SEC ; Indicate no ports available
      RETURN
30$: MOV B L,NSTA(R1),R2 ; Compute the logical tributary #
      SUB R0,R2 ; ...
      MOV B R2,$PBLK+T$SLN+1 ; ...and store in the parameter block
      ASL R2 ; Form a double word index
      ASL R2 ;
      ADD R1,R2 ; Offset into system line table
      MOV B L,MPF(R2),$PBLK+T$SFLG ; Save flags
      MOV L,MPF+S,NMST(R2),$PBLK+T$NMST ; Save network management state
      CLR $PBLK+T$ASN ; No current PDV and channel assignment (Clears C-bit)
      RETURN

```

G.13

MACRO V05.03b Friday 28-Jun-85 22:56 Page 16

```

996                                     .SBTTL  Release a chain of receive data buffers
997                                     ;+
998                                     ;**--RLSCHN-Release a chain of receive data buffers
999                                     ;
1000                                    ;   Return a possible chain of receive data buffers to the pool.
1001                                    ;
1002                                    ; Inputs:
1003                                    ;   R4 = Address of first CCB in the chain
1004
1005 002444 011446  RLSCHN:MOV    (R4),-(SP)      ; Save the link word
1006 002446          CALL    @RDBRT           ; Return this buffer
1007 002452 012604          MOV    (SP)+,R4      ; Recover the link word
1008 002454 001373          BNE    RLSCHN        ; Loop till all returned
1009 002456          RETURN

```

EVL - EVENT LOGGER PROCESS      MACRO V05.03b Friday 28-Jun-85 22:58<sup>F.15</sup>  
Table of contents

|     |     |   |
|-----|-----|---|
| 5-  | 49  | Macro definitions                         |
| 6-  | 65  | DEFINE LOCAL MACROS                       |
| 7-  | 75  | DISPATCH TABLE FOR COMM/EXEC ENTRY        |
| 8-  | 83  | EXEC entry point vector table             |
| 9-  | 116 | LOG NETWORK EVENT                         |
| 10- | 278 | TIMER SERVICE                             |
| 11- | 319 | ALLOCATE BUFFER FOR EVENT                 |
| 12- | 381 | MAP PDV & CHANNEL TO SLN & STATION NUMBER |
| 13- | 408 | MAP PORT NUMBER TO PVC OR DLM NAME        |
| 14- | 476 | QUEUE EVENT TO COLLECTOR TASK             |

```

408 .SBTTL MAP PORT NUMBER TO PVC OR DLM NAME
409
410 *--MAPORT-MAP PORT # TO PVC OR DLM NAME
411
412 MAP THE PORT # SUPPLIED IN THE EVENT RECORD TO A PVC BLOCK OR
413 TO A SLN & STA FOR DATA LINK MAPPING CIRCUITS.
414
415 -
416 INPUTS:
417 R3 - ADDRESS OF EVENT RECORD
418
419 REGISTERS MODIFIED:
420 R2
421
422 MAPORT: SAVMAP ; SAVE CURRENT MAPPING
423 CLR E.PVC(R3) ; ASSUME NOT A VALID PVC
424
425 MOV @PSIPT,R2 ; GET POINTER TO PSI HOME BLOCK
426 BEQ 60$ ; IF EQ, NOT LOADED!!!
427 MOV #H$PTB(R2),-(SP) ; GAIN ACCESS TO THE PORT TABLE
428 CALL @CEACC
429 CLR R2 ; COMPUTE OFFSET IN THE PORT TABLE
430 BISR E.PORT(R3),R2 ; ...
431 ASL R2 ; ...
432 ADD (SP)+,R2 ; COMPUTE ADDRESS OF XCB POINTER
433 MOV (R2),-(SP) ; GAIN ACCESS TO THE XCB
434 CALL @CEACC ; ...
435 MOV (SP)+,R2 ; ...
436 BEQ 60$ ; IF EQ, NO XCB!!!
437
438 TSTB X$USR(R2) ; IS THIS A DATA LINK MAPPING CIRCUIT?
439 BMI 20$ ; IF MI, YES
440
441 BITB #XTPVC,X$TYP(R2)
442 BEQ 60$ ; IF EQ, NOT A PVC
443
444 MOV @PSIPT,R2 ; GET ADDRESS OF PVC LIST HEAD
445 ADD #H$PVC,R2 ; ...
446
447 10$: MOV (R2),E.PVC(R3) ; SAVE ADDRESS OF PVC BLOCK
448 MOV (R2),-(SP) ; GAIN ACCESS TO PVC BLOCK
449 CALL @CEACC
450 MOV (SP)+,R2 ; GET POINTER TO PVC BLOCK
451 BEQ 60$ ; IF EQ, END OF LIST
452 CMPB E.PORT(R3),C$PORT(R2)
453 BNL 10$ ; IF NE, NO MATCH ON THE PORT
454 BR 60$ ; ALL DONE
455
456 20$: MOVB X$USR(R2),R2 ; GET SLN FOR THIS CIRCUIT
457 BIC #C<177>,R2 ; ...
458 MOV R2,E.LIN(R3) ; SAVE IN EVENT RECORD
459 ASL R2 ; FORM WORD OFFSET
460 ADD @SLTMA,R2 ; POINT INTO SYSTEM LINE INDEX TABLE
461 MOV (R2),R2 ; GET ADDRESS OF SYSTEM LINE TABLE
462
463 MAP L.DLM(R2) ; MAP DLC LINE TABLES
464 MOV L.DLS(R2),R2 ; GET POINTER TO DLC LINE TABLE
465 ADD #O$ACHD,R2 ; POINT TO ACTIVE STATION LIST HEAD

```

```

152 .SBTTL Outgoing connect request
153
154 **--SNDCON-Outgoing connect request
155
156 This routine will allocate logical link resources and transmit a
157 connect initiate message.
158
159 Inputs:
160 R5 = Address of database descriptor
161
162 Outputs:
163 'C' Clear - Connect initiate message transmitted
164 'C' Set - Unable to transmit message
165 N$ERRC(R5) - Error code
166
167 Outputs:
168 R3 = Virtual address of LLT
169
170 Registers modified:
171 R0, R1, R2, R4
172
173 000170 SNDCON::CALL MAPNAM ; Map node name to node address
174 000174 103505 BCS 120$ ; If CS, no mapping present
175
176 000176 005065 000016 CLR N$SLA(R5) ; Set up link address for transmission
177 000202 005002 CLR R2 ; Assume normal logical link
178 000204 017704 000000G MOV @DECPT,R4 ; Point to the DEC home block
179 000210 016546 000012 MOV N$SNOD(R5),-(SP) ; Recover destination node address/channel #
180 000214 132767 000001 000000G BITB #1,$FLAGS ; Is it a loopback node?
181 000222 001406 BEQ 10$ ; If EQ, its not a loopback node address
182 000224 012702 001000 MOV #LT.LPL*400,R2 ; Mark this as a loopback logical link
183 000230 016465 000014 000012 MOV D$LNUM(R4),N$SNOD(R5) ; Copy the local node number
184 000236 000405 BR 20$
185
186 000240 021664 000014 10$ CMP (SP),D$LNUM(R4) ; Is this a local logical link?
187 000244 001002 BNE 20$ ; If NE, no
188 000246 012702 000400 MOV #LT.LCL*400,R2 ; Mark this as a local logical link
189
190 000252 20$ CALL ADDLNK ; Try to add a new logical link database
191 000256 012601 MOV (SP)+,R1 ; Recover remote node address/channel #
192 000260 103453 BCS 120$ ; If CS, allocation failure
193 000262 050213 BIS R2,(R3) ; Set up logical link type
194
195 .IF DF N$SLI
196
197 BISB $LTYPE,L.TYP(R3); Set logical link type
198
199 .ENDC
200
201 000264 010163 000006 MOV R1,L.REM(R3) ; and remote node address/channel #
202 000270 112713 000002 MOVB #ST$CIS,(R3) ; Set state to connect initiate sent
203 000274 CALLE RTRSES ; Start round trip timer for CI message
204
205 000304 CALL SAVOPT ; Save optional data in the LLT
206
207 000310 CALL SETREQ ; Set up connect request services
208

```

```

609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624 000000
625 000000 012702 000000G
626 000004 122712 000066
627 000010 103413
628
629 000012 005065 000012
630 000016 012703 000054'
631 000022
632 000026 103404
633 000030 012703 000042'
634 000034
635 000040
636
637 000042 000004
638 000044 001750
639 000046 000144
640 000050 000012
641 000052 000001
642
643 000054 000002
644 000056 024000
645 000060 002000

;+
; **--CHKADD--Check node name for all numerics and convert to binary if
; all numerics
;
; Inputs:
; R5 = ECL's ddb address
;
; Outputs:
; 'c' clear - ASCII node address supplied
; N$SNOD(R5) - binary node address
; 'c' set - Node name is not an ASCII node address
;
; Registers modified:
; R2, R3
;
; .PSECT $HIGH
CHKADD: MOV #SDSNOD,R2 ; Get address of specified node name
; CMPB #'6,(R2) ; Is address in range?
; BLO 10$ ; BR if no (check overflow for first
; ; multiply - ASCBIN will check others)
; CLR N$SNOD(R5) ; Initialize node address
; MOV #CVTARE,R3 ; Get area conversion table
; CALL ASCBIN ; Convert area to binary
; BCS 10$ ; BR if invalid area
; MOV #CVTADD,R3 ; Get address conversion table
; CALL ASCBIN ; Convert address to binary
10$: RETURN
;
; CVTADD: .WORD 4 ; Address conversion table (4 words)
; .WORD 1000.
; .WORD 100.
; .WORD 10.
; .WORD 1
;
; CVTARE: .WORD 2 ; Area conversion table (2 words)
; .WORD 10240.
; .WORD 1024.

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40

```
.TITLE DLXCEX - DLX/CEX interface routines
.IDENT /V05.00/
.ENABL LC
```

```
: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.
```

```
: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.
```

```
: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.
```

```
: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.
```

```
: Module description
```

```
: DLX/CEX interface routines
```

```
: Ident history:
```

```
: 4.00 07-NOV-83
:      DECNET-11M V4.0
:      DECNET-11M-PLUS V2.0
```

```
: 5.00 22-JUL-85
:      DECnet-11M/S V4.2
:      DECnet-11M-Plus V3.0
:      DECnet-Micro/R SX V1.0
```

```

409          .SBTTL  Enable complete processing
410
411      ;+
412      ;**-ENBLC-Enable complete processing
413      ;
414      ;   If the link is in the idle state, issue a start request on the channel
415      ;   otherwise ignore this completion.
416      ;
417      ;-
418      ;Inputs:
419      ;   R1 = Completion status
420      ;   R2 = Error offset for characteristics functions
421      ;   R5 = Address of physical link access block
422
423      ENBLC:  BITB   #LS$IDL,(R5)    ; Is the link idle?
424              BEQ    10$             ; If EQ, no ... must be shutdown in progress
425              CALL   STRCHN         ; Start operation of the channel
426
427      10$:    RETURN
  
```



```

DDDDDDDD LL XX XX CCCCCCCC HH HH RRRRRRRR
DDDDDDDD LL XX XX CCCCCCCC HH HH RRRRRRRR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DD DD LL XX XX CC HH HH RR RR
DDDDDDDD LLLLLLLLLL XX XX CCCCCCCC HH HH RR RR
DDDDDDDD LLLLLLLLLL XX XX CCCCCCCC HH HH RR RR

```

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
          SS          TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS   TT
          SSSSSS   TT
LL          SS       TT
LL          SS       TT
LL          SS       TT
LL          SS       TT
LLLLLLLLLLL SSSSSSSS TT
LLLLLLLLLLL SSSSSSSS TT

```

DLXCHR      CREATED BY    MACRO    ON 28-JUN-85 AT 22:53      PAGE 2      G 6

MACRO CROSS REFERENCE      CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |       |       |       |                  |
|---------|-------|-------|-------|-------|-------|------------------|
| CALL    | 5-88  | 9-249 |       |       |       |                  |
| CCBDF\$ | #4-46 | 4-48  |       |       |       |                  |
| CHRD\$  | #4-46 | 4-49  |       |       |       |                  |
| MAP     | #4-45 | 8-200 | 8-208 |       |       |                  |
| PLADF\$ | #4-46 | 4-50  |       |       |       |                  |
| RESRG   | #4-45 |       |       |       |       |                  |
| RETURN  | 5-95  | 6-139 | 7-173 | 8-219 | 9-256 | 10-289    11-313 |
| SAVRG   | #4-45 |       |       |       |       |                  |

```

283 .SBTTL Assign <Line-id> to another LLC
284
285 ;+
286 **--CTCLASS-Assign <Line-id> to another LLC
287
288 This routine is called to change the <Line-id> and process assignment
289 for a line which is not currently assigned.
290
291 Inputs:
292 R1 = Address of system line table
293 R3 = Address of I/O packet
294 I.PRM+6 - RAD50 name of the target LLC
295
296 Registers modified:
297 R0, R1, R2, R3, R4, R5
298
299 000404 005767 000016G CTCLASS: TST $PBLK+T$ASN ; Is <Line-id> assigned to an LLC?
300 000410 001366 BNE IERSU ; If NE, yes
301 000412 CALL FNDPR ; Find the target process
302 000416 103770 BCS CTLERR ; If CS, no such process
303 000420 CALL FNDPR ; Find a free channel on destination process
304 000424 103765 BCS CTLERR ; If CS, none available
305
306 000426 CALL @CCBGT ; Allocate a CCB
307 000432 103760 BCS IENOD ; If CS, allocation failure
308
309 000434 016702 000010G MOV $PBLK+T$SLN,R2 ; Get SLN and logical tributary
310
311 000440 CTLAS1: CALL ASSIGN ; Modify system database
312
313 000444 012764 000002 000012 MOV #ENBX!LDIS,C.STS(R4)
314
315 000452 112764 000010 000004 CTLO1: MOVB #NX.DON,C.NSP(R4)
316 000460 016764 000016G 000006 MOV $PBLK+T$ASN,C.LIN(R4)
317 000466 012764 001006 000010 MOV #FC.CTL!FS.STR,C.FNC(R4)
318 000474 CALLR CCCBL ; Queue CCB to specified LLC

```

DLXCTL      CREATED BY    MACRO    ON 28-JUN-85 AT 22:54      PAGE 1      G 8

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES   |
|--------|------------|--|
| ASSIGN | = ***** GX | 13-311    20-537   |
| CCBGT  | = ***** GX | 11-170    11-232   |
| CCBRT  | = ***** GX | 23-614   |
| CCPTBL | = 000776 R | 19-504    #19-509  |
| CLACT  | = ***** GX | 22-585   |
| CTLABO | 001142 R   | 20-550    22-583    #23-605  |
| CTLASS | 000404 R   | 8-104    #13-298   |
| CTLAS1 | 000440 R   | 11-238    #13-311  |
| CTLCCP | 000742 RG  | #19-497  |
| CTLDEA | 000532 R   | 8-105    #15-359   |
| CTLDE1 | 001106 R   | 19-512    #21-565  |
| CTLDON | 001160 R   | 19-513    21-568   |
| CTLERR | 000400 R   | 9-132    9-136    22-589    #23-612    11-230    #12-279    13-302    13-304    14-338 |
| CTLLST | 000646 R   | 16-393   |
| CTLOFF | 000500 R   | 8-107    #17-418   |
| CTLOF1 | 000554 R   | 8-102    #14-334   |
| CTLOF2 | 001116 R   | 14-343    #15-367  |
| CTLON  | 000130 R   | 19-511    #22-583  |
| CTLON1 | 000452 R   | 8-101    #10-165   |
| CTLOWN | 000604 R   | 10-176    #13-315  |
| CTLREA | 000232 R   | 8-106    #16-388   |
| CTLRE1 | 001010 R   | 8-103    #11-224   |
| CTLRE2 | 001152 R   | 19-509    #20-534  |
| CTLSER | 000172 R   | 19-510    #23-609  |
| CTLSST | 000654 R   | 8-109    #11-198   |
| C.BUF  | 000014     | 8-108    #17-436   |
| C.BUF2 | 000024     | *18-468    *18-469    19-498    19-499    20-543                                       |
| C.CNT  | 000020     | *11-237    20-538    23-609  |
| C.FLG  | 000022     | *18-470    19-500  |
| C.FNC  | 000010     | *18-471    19-501  |
| C.LIN  | 000006     | *11-242    *13-317    *15-369    *20-544   |
| C.NSP  | 000004     | *11-241    *13-316    *15-368    19-497    *20-543                                     |
| C.STS  | 000012     | *11-240    *13-315    *14-342    *15-365    *18-467    *20-542                         |
| DEASSN | = ***** GX | *10-175    *11-243    *13-313    *15-370    19-503    *20-545    *20-546               |
| ENBX   | = 000002 G | 21-567   |
| FC.CTL | = 000006   | #7-78    13-313    20-546  |
| FINDC  | = ***** GX | 11-242    13-317    15-369   |
| FINDL  | = ***** GX | 8-87    8-88    8-89    8-90    8-91    8-92    8-93                                   |
| FNCTBL | = 000022 R | 8-85    8-86    9-149  |
| FNDFR  | = ***** GX | #8-101    11-229    13-303   |
| FNDPR  | = ***** GX | 11-224    13-301   |
| FS.STP | = 002000   | 11-242    15-369   |
| FS.STR | = 001000   | 13-317    20-544   |
| IEABO  | 000344 R   | 11-205    #12-264  |
| IEALC  | 000360 R   | 10-166    #12-270  |
| IEOD   | 000374 R   | 10-171    11-233    14-335   |
| IERSU  | 000366 R   | #12-273    13-307    14-340    15-363  |
| IEWLK  | 000352 R   | 11-208    #12-267  |
| IE.ABO | = ***** GX | 12-264    23-605   |
| IE.ALC | = ***** GX | 12-270   |
| IE.MOD | = ***** GX | 16-392   |

DLXDAT      CREATED BY MACRO ON 28-JUN-85 AT 22:54      PAGE 1      6 9

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL   | VALUE      | REFERENCES                            |
|----------|------------|---------------------------------------|
| ACHCK    | 000066 RG  | #7-96                                 |
| BLXIO    | 000070 RG  | #7-97                                 |
| CCBGT    | 000104 RG  | #7-115                                |
| CCBRT    | 000106 RG  | #7-116                                |
| CEMUL    | 000110 RG  | #7-117                                |
| CMPDV    | 000112 RG  | #7-118                                |
| FL\$ADR  | = 000001 G | #6-63                                 |
| FL\$PRO  | = 000002 G | #6-64                                 |
| INIBF    | 000072 RG  | #7-98                                 |
| IOFIN    | 000074 RG  | #7-99                                 |
| KISAR6   | = ***** GX | 7-95                                  |
| KSAR6    | 000064 RG  | #7-95                                 |
| LDBGT    | 000114 RG  | #7-119                                |
| LLCRQ    | 000116 RG  | #7-120                                |
| LLCRS    | 000120 RG  | #7-121                                |
| LLCTA    | 000122 RG  | #7-122                                |
| MAXOV    | 000124 RG  | #7-123                                |
| PDVID    | 000126 RG  | #7-124                                |
| PDVTA    | 000130 RG  | #7-125                                |
| PLAST    | 000000 RG  | #8-140                                |
| QUEBF    | 000076 RG  | #7-105                                |
| RDBRT    | 000132 RG  | #7-126                                |
| RDBSZ    | 000134 RG  | #7-127                                |
| RELOC    | 000100 RG  | #7-106                                |
| R\$SMP   | = *****    | 6-75      7-101      7-107      7-112 |
| SLTMA    | 000136 RG  | #7-128                                |
| SLTNM    | 000140 RG  | #7-129                                |
| TSTBF    | 000102 RG  | #7-111                                |
| T\$LEN   | 000024     | 6-52                                  |
| \$ACCESS | 000050 RG  | #6-66                                 |
| \$ACHCK  | = ***** GX | 7-96                                  |
| \$ADR    | 000036 RG  | #6-60                                 |
| \$BIAS   | 000052 RG  | #6-68                                 |
| \$BIGBF  | 000024 RG  | #6-54                                 |
| \$BIGSZ  | 000030 RG  | #6-55                                 |
| \$BLXIO  | = ***** GX | 7-97                                  |
| \$BUFFG  | 000032 RG  | #6-56                                 |
| \$BYTES  | 000056 RG  | #6-70                                 |
| \$CCBGT  | = ***** GX | 7-115                                 |
| \$CCBRT  | = ***** GX | 7-116                                 |
| \$CEMUL  | = ***** GX | 7-117                                 |
| \$CMPDV  | = ***** GX | 7-118                                 |
| \$DLXCH  | 000033 RG  | #6-57                                 |
| \$DLXPD  | 000034 RG  | #6-58                                 |
| \$DLXVL  | = 000027 G | #7-130                                |
| \$DLXVT  | 000062 RG  | #7-93                                 |
| \$FLAGS  | 000046 RG  | #6-62                                 |
| \$INIBF  | = ***** GX | 7-98                                  |
| \$IOFIN  | = ***** GX | 7-99                                  |
| \$IOS    | 000060 RG  | #6-71                                 |
| \$LDBGT  | = ***** GX | 7-119                                 |
| \$LLCRQ  | = ***** GX | 7-120                                 |

DLXLIN      CREATED BY    MACRO    ON 28-JUN-85 AT 22:55      PAGE 1      G 10

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES |
|---------|------------|------------|
| ABORQ   | 000046 RG  | #7-98      |
| CCBGT   | = ***** GX | 6-69       |
| CF.CHN  | = 000001   | 6-74       |
| C.FLG   | 000022     | *6-74      |
| C.FNC   | 000010     | *6-72      |
| C.LIN   | 000006     | *6-73      |
| DISRQ   | 000066 RG  | #7-116     |
| DOFNC   | 000000 RG  | #6-69      |
| FC.CTL  | = 000006   | 7-165      |
| FS.DIS  | = 013000   | 7-172      |
| FS.ENB  | = 012000   | 8-200      |
| FS.MNT  | = 004000   | 8-209      |
| FS.STP  | = 002000   | 7-165      |
| FS.STR  | = 001000   | 8-206      |
| I\$SAS  | = *****    | 5-50       |
| LF\$BRO | = 002000   | 7-99       |
| LF\$CHN | = 004000   | 8-196      |
| LF\$ENB | = 000400   | 7-134      |
| LF\$MOP | = 001000   | 8-207      |
| LF\$RSR | = 100000   | 6-79       |
| LF.ACT  | = 100000   | #5-52      |
| LF.BRO  | = 000400   | #5-52      |
| LF.BWT  | = 000007   | #5-52      |
| LF.ENA  | = 002000   | #5-52      |
| LF.LPB  | = 001000   | #5-52      |
| LF.MDC  | = 000100   | #5-52      |
| LF.MFL  | = 004000   | #5-52      |
| LF.MTP  | = 000020   | #5-52      |
| LF.PAC  | = 000200   | #5-52      |
| LF.RDY  | = 040000   | #5-52      |
| LF.REA  | = 010000   | #5-52      |
| LF.SER  | = 000040   | #5-52      |
| LF.TIM  | = 000010   | #5-52      |
| LF.UNL  | = 020000   | #5-52      |
| LF.X2P  | = 000000   | #5-52      |
| LN.CLO  | = 000000   | #5-52      |
| LN.DUM  | = 000005   | #5-52      |
| LN.LOA  | = 000004   | #5-52      |
| LN.LOO  | = 000003   | #5-52      |
| LN.OAU  | = 000003   | #5-52      |
| LN.OFF  | = 000001   | #5-52      |
| LN.ON   | = 000000   | #5-52      |
| LN.OOP  | = 000004   | #5-52      |
| LN.OPE  | = 000001   | #5-52      |
| LN.REF  | = 000002   | #5-52      |
| LN.SER  | = 000002   | #5-52      |
| LN.STA  | = 000017   | #5-52      |
| LN.SUB  | = 000360   | #5-52      |
| LN.TRI  | = 000006   | #5-52      |
| LR\$DIS | = 000020   | 7-101      |
| LR\$STP | = 000010   | 7-116      |
| LS\$DAT | = 000004   | 7-138      |

DLXQIO - ISSUE I/O TO DLX  
 DLXQIO - ISSUE I/O REQUEST TO NX: MACRO V05.03b Saturday 29-Jun-85 12:23 Page 8

```

91      .SBTTL DLXQIO - ISSUE I/O REQUEST TO NX:
92      ;+
93      ;**DLXQIO-ISSUE I/O REQUEST TO DLX
94      ;
95      ; THIS ROUTINE IS CALLED TO ISSUE A DLX REQUEST.
96      ;
97      ; INPUTS:
98      ; R0 = FUNCTION CODE
99      ; R1 = P1 (OPTIONAL)
100     ; R4 = ADDRESS OF CONTEXT BLOCK
101     ;
102     ; OUTPUTS:
103     ; IF CC, OPERATION SUCCEEDED
104     ; ELSE, OPERATION FAILED, ERROR CODE FILLED INTO DLXMSG STRING
105     ;
106     ; REGISTERS:
107     ; NO REGISTERS MODIFIED
108     ; -
109
110     DLXQIO::
111     000026      CALL    $$SAVAL                ; SAVE ALL REGISTERS
112     000026
113     ;
114     ; FORMAT LINE-ID STRING FOR DLX
115     ;
116     .IF NDF M$$ACP
117
118     000032 010403      MOV    R4,R3                ; COPY THE CONTEXT AREA POINTER
119     000034 062703 000166      ADD    #L$SCR,R3        ; POINT TO THE SCRATCH BUFFER
120
121     .IFF ;NDF M$$ACP
122
123     MOV    CURCTX,R3                ; COPY THE CONTEXT AREA POINTER
124     MOV    R3,R5                    ;
125     ADD    #C$LOC,R3                ; POINT TO THE SCRATCH BUFFER
126     ADD    #C$STAT,R5               ; ... AND THE I/O STATUS BLOCK
127
128     .IFTF ;NDF M$$ACP
129
130     000040      SAVRG    <R3>                ; SAVE R3
131
132     .IF DF S$$BAS ! R$$RTR
133
134     CALL    FMTLN2                  ; FORMAT PHYSICAL LINE-ID
135
136     .IFF
137
138     000042      CALL    FMTLIN                ; FORMAT A LINE-ID STRING
139
140     .ENDC ; DF S$$BAS ! R$$RTR
141
142     000046      RESRG    <R3>                ; RESTORE LINE-ID POINTER
143     000050 112302      MOVB   (R3)+,R2        ; GET STRING LENGTH
144
145     ;
146     ; ISSUE I/O
147     ;
148     .IFT ;NDF M$$ACP

```

DLXSUB - DLX subroutines  
Macro definitions

MACRO V05.03b Friday 28-Jun-85 22:56 Page 5

G 12

```
42 .SBTTL Macro definitions
43
44 .MCALL SAVRG,RESRG,MAP,SAVMAP,RESMAP
45 .MCALL CCBDF$,PDVDF$,PLADF$,SLTDF$,CHRDf$
46
47 000000 CCBDF$ ; Define CCB offsets
48 000000 PDVDF$ ; Define PDV offsets
49 000000 PLADF$ ; Define PLA offsets
50 000000 SLTDF$ ; Define SLT offsets
51 000000 CHRDf$ ; Define characteristics symbols
```



```

535 .SBTTL Validate privileged user and search for <Line-id>
536
537 +
538 **FINDC-Validate privileged user and search for <Line-id>
539
540 Validate that this request comes from a privileged task and then
541 search for the specified <Line-id>.
542
543 -
544 Inputs:
545 R2 = Dynamic assignment flag
546       0 - Dynamically assign broadcast ports
547       <>0 - Do not dynamically assign broadcast ports
548 R3 = Address of I/O packet
549       I.PRM - Bias of <Line-id> string
550       I.PRM+2 - Virtual address of <Line-id> string
551       I.PRM+4 - Length of <Line-id> string
552
553 Outputs:
554 R0 = Error code (if applicable)
555       IE.PRI - Privilege violation
556       IE.NSF - No such <Line-id> in system
557       IE.OFL - Device offline (RSX-11M-Plus only)
558 R1 = Address of system line table
559 'C' Clear - <Line-id> successfully found
560 'Z' Clear - <Line-id> is assigned
561 'Z' Set - No current assignment
562 'C' Set - Error (R0 contains error code)
563
564 Registers modified:
565 R0
566
567 .ENABL LSB
568
569 FINDC:: MOV I.TCB(R3),R0 ; Get TCB address of issuing task
570          BIT #I3.PRIV,I.ST3(R0)
571          BEQ 30$ ; If EQ, issuing task is not privileged
572
573 +
574 **FINDL-Search for <Line-id>
575
576 Search the COMM/EXEC databases for the specified <Line-id>.
577
578 -
579 Inputs:
580 R2 = Dynamic assignment flag
581       0 - Dynamically assign broadcast ports
582       <>0 - Do not dynamically assign broadcast ports
583 R3 = Address of I/O packet
584       I.PRM - Bias of <Line-id> string
585       I.PRM+2 - Virtual address of <Line-id> string
586       I.PRM+4 - Length of <Line-id> string
587
588 Outputs:
589 R0 = Error code (if applicable)
590       IE.PRI - Privilege violation
591       IE.NSF - No such <Line-id> in system
592       IE.OFL - Device offline (RSX-11M-Plus only)
593 R1 = Address of system line table
594 'C' Clear - <Line-id> successfully found

```

```

1011      .SBTTL  Compute address of physical link access block
1012      ;+
1013      ;*-STLNK-Compute address of physical link access block
1014      ;*-STLNA-Alternate entry
1015      ;
1016      ;   Compute the address of the physical link access block given a channel
1017      ;   number.
1018      ;
1019      ; Inputs:
1020      ;   R0 = Channel # (STLNA only)
1021      ;   R4 = Address of a CCB with a channel # in C.LIN (STLNK only)
1022      ;
1023      ; Outputs:
1024      ;   R5 = Address of the physical link access block
1025      ;
1026      ; Registers modified:
1027      ;   R0, R1
1028      ;
1029      STLNK:: MOVB  C.LIN(R4),R0      ; Extract the channel #
1030
1031      STLNA:: BIC   #^C<377>,R0      ; Isolate the channel #
1032      MOVB  R0,-(SP)                ; and save for later
1033
1034      MOV    #L$LEN,R1              ; Set length of a physical access block
1035      CALL  @CEMUL                   ; Compute offset
1036      MOV    R1,R5                  ; Compute address in the table
1037      ADD    #PLAST,R5
1038      MOVB   (SP)+,L$LIN(R5)        ; Initialise the channel #
1039      RETURN
  
```

|      |        |        |         |
|------|--------|--------|---------|
| 1029 | 002460 | 115400 | 000006  |
| 1030 |        |        |         |
| 1031 | 002464 | 042700 | 177400  |
| 1032 | 002470 | 110046 |         |
| 1033 |        |        |         |
| 1034 | 002472 | 012701 | 000026  |
| 1035 | 002476 |        |         |
| 1036 | 002502 | 010105 |         |
| 1037 | 002504 | 062705 | 000000G |
| 1038 | 002510 | 112665 | 000006  |
| 1039 | 002514 |        |         |

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47

.TITLE EVL - EVENT LOGGER PROCESS  
.IDENT /V05.00/

.. COPYRIGHT (C) 1980, 1981, 1982, 1983, 1985 BY  
.. DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

.. THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A  
.. SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE  
.. INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR  
.. ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE  
.. MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH  
.. SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE  
.. TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN  
.. IN DEC.

.. THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT  
.. NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL  
.. EQUIPMENT CORPORATION.

.. DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF  
.. ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

.. MODULE DESCRIPTION

.. DRIVER INTERFACE TO EVENT LOGGER PROCESS

.. DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

.. IDENT HISTORY:

- .. 1.00 31-JAN-80  
.. 2.00 16-APR-82  
..     DECNET-11M V3.1  
..     DECNET-11M-PLUS V1.1  
.. 4.00 07-NOV-83  
..     DECNET-11M V4.0  
..     DECNET-11M-PLUS V2.0  
.. 5.00 22-JUL-85  
..     DECnet-11M/S V4.2  
..     DECnet-11M-Plus V3.0  
..     DECnet-Micro/RSX V1.0

```

465
466 001074 011202          30$:  MOV      (R2),R2      ; GET POINTER TO NEXT STATION TABLE
467 001076 001412          60$:  BEQ      60$,      ; IF EQ, NO MORE
468 001100 126362 000040 000003  CMPB    E,PORT(R3),Q$PORT(R2)
469 001106 001372          30$:  BNE      30$,      ; IF NE, NO MATCH ON PORT
470 001110 116262 000002 (J0025  MOVB    Q$STN(R2),E,LIN+1(R3)
471 001116 142763 000200 000020  BICB    #EV.PRT,E.CTL(R3)
472
473 001124          60$:  RESMAP      ; RESTORE MAPPING
474 001130          RETURN

```

|     |        |        |         |         |                    |  |
|-----|--------|--------|---------|---------|--------------------|--|
| 209 | 000314 | 004167 | 000000G | JSR     | R1,GETLDB          | ; Allocate a large data buffer             |
| 210 | 000320 | 000    |         | .BYTE   | NT\$CON            | ; This is a connect message                |
| 211 | 000321 | 030    |         | .BYTE   | MF,CTL!MC.CI       | ; Connect message flags                    |
| 212 | 000322 | 103430 |         | BCS     | 110\$              | ; If CS, allocation failure                |
| 213 |        |        |         |         |                    |  |
| 214 | 000324 | 116722 | 000000G | MOVB    | \$SRVCS,(R2)+      | ; Fill in requested services               |
| 215 | 000330 | 112722 | 000002  | MOVB    | #CV\$40,(R2)+      | ; Fill in info field - NSP version 4.0     |
| 216 | 000334 | 017701 | 000000G | MOV     | @DECPTR1           | ; Point to the DEC home block              |
| 217 | 000340 | 116122 | 000036  | MOVB    | D\$SEG(R1),(R2)+   | ; Fill in segment size                     |
| 218 | 000344 | 116122 | 000037  | MOVB    | D\$SEG+1(R1),(R2)+ | ; .....                                    |
| 219 | 000350 |        |         | CALL    | BLDSES             | ; Build session control portion of message |
| 220 | 000354 | 103411 |         | BCS     | 100\$              | ; If CS, format error                      |
| 221 |        |        |         |         |                    |  |
| 222 | 000356 |        |         | COUNT\$ | E\$NCS             | ; Count connect requests sent              |
| 223 | 000364 |        |         | CALLE   | SNSESD             | ; Transmit the CI message                  |
| 224 | 000374 | 000241 |         | CLC     |                    | ; Indicate success                         |
| 225 | 000376 |        |         | RETURN  |                    |  |
| 226 |        |        |         |         |                    |  |
| 227 | 000400 |        |         | 100\$:  | CALL @LDBRT        | ; Release the large data buffer            |
| 228 | 000404 |        |         | 110\$:  | CALL REMLNK        | ; Remove the link database                 |
| 229 | 000410 |        |         | 120\$:  | RETURN             |  |

```

647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664 000062
665 000062
666 000064 012304
667 000066 012301
668 000070 005000
669 000072 152200
670 000074 162700 000060
671 000100 002413
672 000102 022700 000011
673 000106 103410
674
675
676
677
678
679
680
681 000110
682
683
684
685 000114 060165 000012
686 000120 103403
687 000122
688 000126 005727
689 000130 000261
690 000132
691 000134
692
693 000001

;+
;**-ASCBIN-Convert ASCII to binary and accumulate result
;
Inputs:
R2 = Next character in node name string
R3 = Address of conversion table
R5 = Address of ECL's ddb
;
Outputs:
'C' clear - valid conversion
      NSSNOD(R5) - updated binary node address
      R2 = address of next byte in node name string
'C' set - Node name is not an ASCII node address
;
Registers modified:
R0, R3, R4
;
ASCBIN: .PSECT $HIGH
        SAVRG <R1> ; Save a register
        MOV (R3)+,R4 ; Get number of bytes to convert
10$:    MOV (R3)+,R1 ; Get multiplicand
        CLR R0 ; Get next character
        BISB (R2)+,R0 ;
        SUB #60,R0 ; Convert to binary
        BLT 20$ ; BR if non-numeric
        CMP #9,R0 ; Is it 0-9?
        BLO 20$ ; BR if no
;
        .IF DF R$$EIS
        MUL R1,R0 ; Accumulate the result
        .IFF
        CALL @CEMUL ; Accumulate the result
        .ENDC
;
        ADD R1,NSSNOD(R5) ; Update binary node address
        BCS 20$ ; BR if overflow
        SOB R4,10$ ; Convert all digits
        TST (PC)+ ; Indicate valid conversion
20$:    SEC ; Indicate non-numeric node address
        RESRG <R1> ; Restore register
        RETURN
        .END

```

```
42                                     .SBTTL  Macro definitions
43
44                                     .MCALL  SAVRG,RESRG,MAP
45                                     .MCALL  CCBDF$,PDVDF$,PLADF$,SLTDF$,DLXDF$
46
47 000000 CCBDF$                      ; Define CCB offsets
48 000000 PDVDF$                      ; Define PDV offsets
49 000000 PLADF$                      ; Define PLA offsets
50 000000 SLTDF$                      ; Define SLT offsets
51 000000 DLXDF$                      ; Define DLX symbols
```

```

427                                     .SBTTL Start complete processing
428
429
430
431
432
433
434
435
436
437
438
439 001054 132715 000002
440 001060 001426
441
442 001062 032715 002000
443
444
445 001066 001402
446 001070 052715 004000
447
448 001074
449 001102 103411
450 001104 012700 000000C
451 001110 032715 002000
452 001114 001402
453 001116 052700 000400
454 001122
455
456 001126 112715 000004
457 001132
458
459 001136

                                     :+
                                     **-STRTC-Start complete processing
                                     :
                                     If the link is in the starting state, complete the outstanding control
                                     request and initiate data transmission.
                                     :-
                                     Inputs:
                                     R1 = Completion status
                                     R2 = Error offset for characteristics functions
                                     R5 = Address of physical link access block
STRTC: BITB    #LS$STR,(R5)    ; Is the link starting?
        BEQ     40$           ; If EQ, no ... must be shutdown in progress
                                     :
        BIT     #LF$BRO,(R5)   ; For the moment only support chaining
                                     ; on broadcast devices - later change to
                                     ; device/DLC dependant support
        BEQ     10$           ; If EQ, non-broadcast device
        BIS     #LF$CHN,(R5)  ; Indicate DLC supports chaining
                                     :
10$:    $DQPKT  FNC           ; Dequeue the start request
        BCS     30$           ; If CS, none present
        MOV     #IS,SUC&377,R0 ; Assume non-broadcast channel
        BIT     #LF$BRO,(R5)  ; Is this a broadcast link?
        BEQ     20$           ; If EQ, no
        BIS     #<XV.BRO*400>,R0 ; Tell user this is a broadcast channel
20$:    CALL    RQALT         ; Complete the start function
                                     :
30$:    MOVB    #LS$DAT,(R5)  ; Set state to data
        CALLR   ITRANS        ; Initiate data transmission
                                     :
40$:    RETURN

```



DLXCHR - DLX Auxilliary charact MACRO V05.03b Friday 28-Jun-85 22:53<sup>H 5</sup>  
Table of contents

|     |     |   |
|-----|-----|---|
| 5-  | 52  | Process receive auxilliary characteristics  |
| 6-  | 105 | Retrieve Ethernet address                   |
| 7-  | 141 | Retrieve Ethernet protocol                  |
| 8-  | 175 | Retrieve destination Ethernet address       |
| 9-  | 221 | Process transmit auxilliary characteristics |
| 10- | 265 | Define Ethernet address for transmit        |
| 11- | 291 | Define Ethernet protocol for transmit       |

```

LL          SSSSSSSS  TTTT TTTT TTTT
LL          SSSSSSSS  TTTT TTTT TTTT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LL          SSSSSS     TT
LL          SSSSSS     TT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LL          SS         TT
LLLLLLLLLLL SSSSSSSS  TT
LLLLLLLLLLL SSSSSSSS  TT

```

```

320 .SBTTL Turn <Line-id> off
321
322 :+
323 :*-CTLOFF-Turn <Line-id> off
324 :
325 :   This routine is called to stop the operation of the LLC process
326 :   assigned to the <Line-id>.
327 :-
328 :   Inputs:
329 :       R1 = Address of system line table
330 :       R3 = Address of I/O packet
331 :
332 :   Registers modified:
333 :       R0, R1, R2, R3, R4, R5
334 000500 005767 000016G CTLOFF: TST    $PBLK+T$ASN    ; Is <Line-id> assigned to an LLC?
335 000504 001725          BEQ      IEALC      ; If EQ, no
336
337 000506          CALL    PRCLD      ; Is the target process loaded?
338 000512 103732        BCS      CTLERR    ; If CS, no
339 000514          CALL    @CCBGT     ; Allocate a CCB
340 000520 103725        BCS      IENOD     ; If CS, allocation failure
341
342 000522 112764 000004 000004 MOVB    #NX.OFF,C.NSP(R4)
343 000530 000411          BR      CTLOFF1  ; Enter common code

```

DLXCTL      CREATED BY MACRO ON 28-JUN-85 AT 22:54      PAGE 2      H 8  
 SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL | VALUE      | REFERENCES                                  |
|--------|------------|---|
| IE.NOD | = ***** GX | 12-276                                      |
| IE.ONP | = ***** GX | 9-130                                       |
| IE.RSU | = ***** GX | 12-273                                      |
| IE.WLK | = ***** GX | 12-267                                      |
| INIBF  | = ***** GX | 9-144                                       |
| ISALT  | = 000336 R | 11-201      11-228      #12-261      15-360 |
| ISSUC  | = ***** GX | 16-399      17-445                          |
| IS.SET | = ***** GX | 12-261                                      |
| IS.SUC | = ***** GX | 23-613                                      |
| ISSAS  | = *****    | 5-50  |
| I.FCN  | = ***** GX | 9-127      *9-143                           |
| I.PRM  | = ***** GX | 11-207      16-388      17-441              |
| LDIS   | = 000000   | #7-76      13-313                           |
| LENB   | = 000001   | G      20-545                               |
| LF.ACT | = 100000   | #5-52                                       |
| LF.BRO | = 000400   | #5-52                                       |
| LF.BWT | = 000007   | #5-52                                       |
| LF.ENA | = 002000   | #5-52                                       |
| LF.LPB | = 001000   | #5-52                                       |
| LF.MDC | = 000100   | #5-52                                       |
| LF.MFL | = 004000   | #5-52                                       |
| LF.MTP | = 000020   | #5-52                                       |
| LF.PAC | = 000200   | #5-52                                       |
| LF.RDY | = 040000   | #5-52                                       |
| LF.REA | = 010000   | #5-52                                       |
| LF.SER | = 000040   | #5-52                                       |
| LF.TIM | = 000010   | #5-52                                       |
| LF.UNL | = 020000   | #5-52                                       |
| LF.X2P | = 000000   | #5-52                                       |
| LLCRS  | = ***** GX | 18-472                                      |
| LN.CLO | = 000000   | #5-52                                       |
| LN.DUM | = 000005   | #5-52                                       |
| LN.LOA | = 000004   | #5-52                                       |
| LN.LOO | = 000003   | #5-52                                       |
| LN.OAU | = 000003   | #5-52      11-202                           |
| LN.OFF | = 000001   | #5-52      22-588                           |
| LN.ON  | = 000000   | #5-52      11-204                           |
| LN.OOP | = 000004   | #5-52                                       |
| LN.OPE | = 000001   | #5-52                                       |
| LN.REF | = 000002   | #5-52                                       |
| LN.SER | = 000002   | #5-52      11-200                           |
| LN.STA | = 000017   | #5-52      11-199      17-418      22-587   |
| LN.SUB | = 000360   | #5-52      17-436                           |
| LN.TRI | = 000006   | #5-52                                       |
| L.COST | = 000015   | #5-52                                       |
| L.CTL  | = 000012   | #5-52                                       |
| L.CVA  | = 177776   | #5-52                                       |
| L.DDM  | = 000002   | #5-52                                       |
| L.DDS  | = 000004   | #5-52                                       |
| L.DLC  | = 000003   | #5-52                                       |
| L.DLM  | = 000006   | #5-52                                       |
| L.DLS  | = 000010   | #5-52                                       |

DLXDAT      CREATED BY MACRO ON 28-JUN-85 AT 22:54      PAGE 2      H 9

SYMBOL CROSS REFERENCE      CREF 04.00

SYMBOL    VALUE                    REFERENCES

|         |            |       |
|---------|------------|-------|
| \$LLCRS | = ***** GX | 7-121 |
| \$LLCTA | = ***** GX | 7-122 |
| \$MAXOV | = ***** GX | 7-123 |
| \$PBLK  | 000000 RG  | #6-52 |
| \$PDVID | = ***** GX | 7-124 |
| \$PDVTA | = ***** GX | 7-125 |
| \$PRO   | 000044 RG  | #6-61 |
| \$QUEBF | = ***** GX | 7-105 |
| \$RDBRT | = ***** GX | 7-126 |
| \$RDBSZ | = ***** GX | 7-127 |
| \$RELOC | = ***** GX | 7-106 |
| \$SLTMA | = ***** GX | 7-128 |
| \$SLTNM | = ***** GX | 7-129 |
| \$STBFB | = ***** GX | 7-111 |
| \$VA    | 000054 RG  | #6-69 |

DLXLIN      CREATED BY    MACRO    ON 28-JUN-85 AT 22:55      PAGE 2      H 10  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE    | REFERENCES |
|---------|----------|------------|
| LS\$ERR | = 000200 | 7-132      |
| LS\$RST | = 000020 | 7-132      |
| LS\$SIP | = 000010 | 7-98       |
| LS\$STR | = 000002 | 7-138      |
| L\$LIN  | 000006   | 6-73       |
| L\$RSR  | 000002   | *6-75      |
| L\$TIP  | 000003   | 7-147      |
| L.COST  | 000015   | #5-52      |
| L.CTL   | 000012   | #5-52      |
| L.CVA   | 177776   | #5-52      |
| L.DDM   | 000002   | #5-52      |
| L.DDS   | 000004   | #5-52      |
| L.DLC   | 000003   | #5-52      |
| L.DLM   | 000006   | #5-52      |
| L.DLS   | 000010   | #5-52      |
| L.FLG   | 000000   | #5-52      |
| L.KRBA  | 000016   | #5-52      |
| L.LEN   | = 000022 | #5-52      |
| L.MPF   | 000022   | #5-52      |
| L.NMST  | 000020   | #5-52      |
| L.NSTA  | 000014   | #5-52      |
| L.OWNR  | 000021   | #5-52      |
| L.UNT   | 000013   | #5-52      |
| N\$SVCT | = *****  | 6-76       |
| RCVAB   | = *****  | GX 7-146   |
| R\$11D  | = *****  | 5-50       |
| R\$11M  | = 000000 | 5-50       |
| R\$11S  | = *****  | 5-50       |
| SF.ACT  | = 000200 | #5-52      |
| SF.ENA  | = 000100 | #5-52      |
| SF.LPB  | = 000004 | #5-52      |
| SF.MFL  | = 000040 | #5-52      |
| SF.PAC  | = 000020 | #5-52      |
| SF.REA  | = 000010 | #5-52      |
| SF.SER  | = 000001 | #5-52      |
| SF.SVC  | = 000002 | #5-52      |
| SF.UNL  | = 000040 | #5-52      |
| STOPC   | = *****  | GX 7-180   |
| STPCHN  | 000156   | RG 7-163   |
| STPRQ   | 000074   | RG 7-131   |
| STRCHN  | 000250   | RG 7-196   |
| S.COST  | 000001   | #5-52      |
| S.FLG   | 000000   | #5-52      |
| S.LEN   | 000004   | #5-52      |
| S.NMST  | 000002   | #5-52      |
| S.OWNR  | 000003   | #5-52      |
| XMTAB   | = *****  | GX 7-149   |
| X\$MCB  | = *****  | 5-50       |
| ZF.COU  | = 001000 | 5-50       |
| ZF.DDM  | = 000001 | 5-50       |
| ZF.DIA  | = 004000 | 5-50       |
| ZF.DLC  | = 000002 | 5-50       |

```

148
149 000052          10$: QIOW$$ R0,#$TMLUN,$$TMEFN,,#IOSB,,<R3,R2,R1> ; ISSUE I/O
150
151                .IFF      ;NDF M$$ACP
152
153          10$: QIO$$ R0,$$TMLUN,,,R5,$DLXAST,<R3,R2,R1> ; ISSUE I/O
154
155                .IFTF     ;NDF M$$ACP
156
157 000116 103010      BCC      20$                ; IF CC, DIRECTIVE SUCCEEDED
158 000120 026727 000000G 000000G      CMP      $$$W,$$IE.UPN      ; ELSE WAS IT A RESOURCE ERROR ?
159 000126 001023      BNE      40$                ; IF NE, NO - REAL ERROR
160 000130      WSIG$$      BR      10$                ; ELSE, WAIT AROUND A WHILE
161 000136 000745      20$:          ; AND TRY IT AGAIN
162 000140
163
164                .IFT      ;NDF M$$ACP
165 000140 116701 177634      MOVB      IOSB,R1                ; GET STATUS RETURN
166
167
168                .IFF      ;NDF M$$ACP
169
170      CALL      FORK                ; FORK AND WAIT FOR QIO COMPLETION
171      MOVB      STATUS,R1          ; GET THE STATUS RETURN
172
173      .ENDC     ;NDF M$$ACP
174
175 000144 003014      BGT      40$                ; IF GT, SUCCESS
176 000146 012700 000020'      MOV      #DLXERR,R0      ; POINT TO ASCII ERROR CODE BUFFER
177 000152 005002      CLR      R2                ; ENABLE ZERO SUPPRESSION
178 000154      CALL      $CBDSG      ; CONVERT IT TO SIGNED DECIMAL
179 000160 012701 000005'      MOV      #DLXTXT,R1      ; POINT TO ERROR TEXT
180 000164 160100      SUB      R1,R0                ; CALC STRING LENGTH
181 000166 110041      MOVB      R0,-(R1)          ; SET STRING LENGTH
182 000170 010164 000056      MOV      R1,$MSG(R4)      ; SET ERROR MESSAGE ADDRESS
183 000174 000261      SEC
184 000176          40$: RETURN      ; ELSE, NO

```

```
53 .SBTTL Assign line to an LLC process
54
55 *
56 **--ASSIGN-Assign line to an LLC process
57
58 The COMM/EXEC databases are updated to reflect the new line assignment
59 and the system line and/or tributary are marked active.
60
61 Inputs:
62 R0 = Target PDV and Channel
63 R1 = Address of channel entry in target PDV
64 R2 = SLN and logical tributary # or the <Line-id>
65 Bit 15=0 - Set LF.ACT and network management state to 'ON'
66 Bit 15=1 - Leave LF.ACT and network management state alone
67
68 .ENABL LSB
69
70 000000 004567 000522 ASSIGN::JSR R5,COASN ; Compute address in reverse mapping table
71
72 MOV R0,(R5) ; Store new PDV and Channel in reverse mapping table
73 MOV R2,(R1) ; Store new SLN and tributary in channel mapping table
74 BIC #ZS.ASN,(R1) ; Make sure channel is no longer assignable
75 TST R2 ; Should we change LF.ACT etc?
76 BMI 20$ ; If MI, no
77
78 *
79 **--STACT-Set system line and/or tributary active
80
81 Mark the system line and/or the tributary active.
82
83 Inputs:
84 R2 = SLN and logical tributary # of the <Line-id>
85 Bit 15=0 - Set network management state to 'ON'
86 Bit 15=1 - Leave network management state alone
87
88 Registers modified:
89 R5
90
91 000020 STACT:: SAVRG <R2> ; Save SLN and tributary #
92 000022 CALL STSLT ; Compute address of the SLT
93
94 000026 052715 100000 BIS #LF.ACT,(R5) ; Mark the line active
95 000032 105765 000014 TSTB L.NSTA(R5) ; Is this a multipoint master or broadcast?
96 000036 001411 BEQ 10$ ; If EQ, no
97
98 000040 000302 SWAB R2 ; Isolate tributary #
99 000042 042702 177700 BIC #^C<77>,R2 ; ...
100 000046 006302 ASL R2 ; Form double word offset
101 000050 006302 ASL R2 ; ...
102 000052 060205 ADD R2,R5 ; Compute address in SLT
103 000054 152765 000200 000022 BISB #SF.ACT,L.MPF(R5)
104
105 000062 10$: RESRG <R2> ; Recover SLN and tributary #
106 000064 100406 BMI 20$ ; If MI, leave network management state alone
107
108 000066 CALL STNMST ; Compute address of network management state
109 000072 142715 000017 BICB #LN.STA,(R5) ; Clear out current state
110 000076 152715 000000 BISB #LN.ON,(R5) ; Set new state to 'ON'
```



```

592      ; 'Z' Clear - <Line-id> is assigned
593      ; 'Z' Set - No current assignment
594      ; 'C' Set - Error (R0 contains error code)
595      ;
596      ; Registers modified:
597      ; R0
598
599 001262 010267 000000G FINDL:: MOV R2,$ACCESS ; Save dynamic assignment flag
600 001266      CALL FNDLN ; Search databases for <Line-id>
601 001272 103422      BCS 40$ ; If CS, not found
602
603 001274 132767 000001 000006G BITB #ZF.DDM,$PBLK+T$FLG
604 001302 001416      BEQ 40$ ; Was this really a <Line-id>?
605 001304 032767 040000 000012G BIT #LF.RDY,$PBLK+T$SLF
606 001312 001412      BEQ 40$ ; Make sure <Line-id> is ready
607 001314 032767 020000 000012G BIT #LF.UNL,$PBLK+T$SLF
608 001322 001006      BNE 40$ ; and not waiting for unload
609
610      .IF DF R$$MPL
611
612      MOV $PBLK+T$KRB,R0 ; Get pointer to KRB
613      BIT #KS.OFL,K.STS(R0)
614      BNE 20$ ; If NE, controller is offline
615      TSTB @NCPU ; Is this a multi-processor system?
616      BEQ 10$ ; If EQ, no
617      BIT K.URM(R0),@URMST ; Is the bus currently online?
618      BEQ 20$ ; If EQ, no
619
620      10$:
621      .ENDC
622 001324 005767 000016G TST $PBLK+T$ASN ; Set condition code according to assignment
623 001330      RETURN
624
625      .IF DF R$$MPL
626
627      20$:
628      MOV #IE.OFL&377,R0 ; Set device offline
629      BR 50$
630
631      .ENDC
632 001332 012700 000000C 30$: MOV #IE.PRI&377,R0 ; Set privilege violation
633 001336 000402      BR 50$
634
635 001340 012700 000000C 40$: MOV #IE.NSF&377,R0 ; Set no such <Line-id>
636 001344 000261      SEC ; Indicate error
637 001346      RETURN
638
639      .DSABL LSB

```

```

1041 .SBTTL Compute address of network management state byte
1042
1043 ;**--STNMST-Compute address of network management state byte
1044 ;
1045 ; Compute the address of the network management state byte for the
1046 ; specified <Line-id>.
1047 ;
1048 ; Inputs:
1049 ; R2 = SLN and logical tributary #
1050 ;
1051 ; Outputs:
1052 ; R5 = Address of the network management state byte
1053 ;
1054 STNMST::SAVRG <R2> ; Save SLN and logical tributary #
1055 CALL STSLT ; Compute address of the SLT
1056 ADD #L.NMST,R5 ; Assume line is not multipoint
1057
1058 002516 032765 000400 177760 BIT #LF.BRO,L.FLG-L.NMST(R5) ; Is line a broadcast line?
1059 002520 001007 BNE 5$ ; BR if yes
1060 002540 032765 000020 177760 BIT #LF.MTP,L.FLG-L.NMST(R5)
1061 002546 001413 BEQ 10$ ; If EQ, line is not multipoint
1062 002550 105765 177774 TSTB L.NSTA-L.NMST(R5)
1063 002554 001410 BEQ 10$ ; If EQ, line is not a multipoint master
1064
1065 002556 000302 5$: SWAB R2 ; Isolate tributary
1066 002560 042702 BIC #^C<77>,R2 ; ...
1067 002564 006302 ASL R2 ; Form double word offset
1068 002566 006302 ASL R2 ; ...
1069 002570 060205 ADD R2,R5 ; Compute address in tributary extension
1070 002572 062705 000004 ADD #L.MPF-L.NMST+S.NMST,R5
1071
1072 002576 10$: RESRG <R2> ; Restore SLN and logical tributary #
1073 002600 RETURN

```

```

49      .SBTTL  Macro definitions
50
51      .MCALL  SLTDF$,EVLDF$,PDVDF$,CCBDF$,INHIB$,ENABL$,SAVRG,RESRG,MAP
52      .MCALL  SAVMAP,RESMAP
53
54      .MCALL  PHBDF$,XCBDF$,PVCDF$,DLMDF$
55
56      000000      PHBDF$      ; DEFINE PSI HOME BLOCK OFFSETS
57      000000      XCBDF$      ; DEFINE XCB OFFSETS
58      000000      PVCDF$      ; DEFINE PVC OFFSETS
59      000000      DLMDF$      ; DEFINE DLM LINE TABLE OFFSETS
60      000000      SLTDF$      ; DEFINE SYSTEM LINE TABLE OFFSETS
61      000000      EVLDF$      ; DEFINE EVENT LOGGER OFFSETS
62      000000      PDVDF$      ; DEFINE PDV OFFSETS
63      000000      CCBDF$      ; DEFINE CCB OFFSETS

```

EVL - EVENT LOGGER PROCESS  
QUEUE EVENT TO COLLECTOR TASK

MACRO V05.03b Friday 28-Jun-85 22:58 Page 14

H 16

476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491 001132  
492 001136 010301  
493 001140 062701 000004  
494 001144 016700 176642  
495 001150 162700 000014  
496  
497 000006  
498  
499  
500  
501 001170 005013  
502 001172 010375 000002  
503 001176 010365 000002  
504 001202  
505 001206  
506  
507 000001

```
.SBTTL QUEUE EVENT TO COLLECTOR TASK
+
**--QUEUE-QUEUE EVENT TO THE COLLECTOR TASK
:
:   TIME STAMP THE EVENT RECORD AND QUEUE IT TO THE COLLECTOR TASK.
:   NOTE THAT THIS ROUTINE ASSUMES THAT IT IS INTERLOCKED WITH THE
:   DEVICE DRIVERS.
-
: INPUTS:
:   R3 - ADDRESS OF EVENT RECORD
:   R5 - POINTER TO EVENT LOGGER PROCESS DATABASE
:
: REGISTERS MODIFIED:
:   R2, R3
:
: QUEUE: SAVRG <R0,R1>      ::: GET SOME FREE REGISTERS
:        MOV R3,R1          ::: COPY ADDRESS OF EVENT RECORD
:        ADD #E.TIME,R1     ::: AND POINT TO TIME-STAMP AREA
:        MOV ITNS,R0        ::: POINT TO CELLS IN EXEC
:        SUB #14,R0         ::: ...
:
:        .REPT 6
:        MOV (R0)+,(R1)+    ::: TIME-STAMP THE EVENT
:        .ENDR
:
:        CLR (R3)           ::: CLEAR OUT LINK WORD
:        MOV R3,@$EVTS+2(R5) ::: QUEUE EVENT TO THE COLLECTOR TASK
:        MOV R3,$$EVTS+2(R5) ::: ...
:        RESRG <R1,R0>     ::: RESTORE REGISTERS
:        RETURN
:
: .END
```

```

231 .SBTTL Build session control portion of connect initiate message
232
233 **BLDSES-Build session control portion of connect initiate message
234
235 Build the session control portion of the connect initiate message
236 indicating the source and destination process indentifications.
237
238 Inputs:
239 R2 = Pointer to next available byte in the message buffer
240 R3 = Virtual address of LLT
241 R4 = Address of CCB
242 R5 = Address of database descriptor
243
244 Outputs:
245 R2 = Pointer past last byte used in message buffer
246 'C' Clear - Valid message built
247 'C' Set - Format error
248 N$ERRC(R5) - ER$FMT
249
250 Registers modified:
251 R0, R1
252
253 BLDSES: SAVRG <R5> ; Get a free register
254 MOV #ER$FMT,N$ERRC(R5)
255
256 MOV #DSDFM,R0 ; Point to destination descriptor
257 CALL BLDDSC ; Build destination process/task descriptor
258 BCS 100$ ; If CS, format error
259
260 MOV #SSRDFM,R0 ; Point to source descriptor
261 CALL BLDDSC ; Build source process/task descriptor
262 BCS 100$ ; If CS, format error
263
264 MOVB #CL,MU1:CL,MU2,(R2)+
265 MOV #SREQID,R0
266 JSR R3,BLDIMG ; Build image mode field
267 .WORD 16 ; Max length = 16. bytes
268 BCS 100$ ; If CS, format error
269
270 MOV #SPASSW,R0
271 JSR R3,BLDIMG ; Build image mode field
272 .WORD 8 ; Max length = 8. bytes
273 BCS 100$ ; If CS, format error
274
275 MOV #SACNT,R0
276 JSR R3,BLDIMG ; Build image mode field
277 .WORD 16 ; Max length = 16. bytes
278 BCS 100$ ; If CS, format error
279
280 MOV (SP),R5 ; Recover address of database descriptor
281 CALL ADDOPT ; Add optional data to message
282 CLC ; Indicate success
283
284 100$: RESRG <R5>
285 RETURN

```

|                  |                  |                   |                 |                  |
|------------------|------------------|-------------------|-----------------|------------------|
| ACCLLT= ***** GX | CE.ERR= 100370   | CS.HFE= 002000    | D\$AMXC 000072  | ESNBS 000020     |
| ADDLNK= ***** GX | CE.ILN= 100350   | CS.LST= 040000    | D\$AMXH 000074  | ESNCR 000034     |
| ADDMAI= ***** GX | CE.LTO= 100356   | CS.MTL= 004000    | D\$ANN 000000   | ESNCS 000036     |
| ADDOPT= ***** GX | CE.MOP= 100372   | CS.RNG= 000010    | D\$BRPR 000102  | ESNIC 000044     |
| ASCBIN 000062R   | CE.NTE= 100361   | CS.ROV= 000004    | D\$BRM 000100   | ESNLN 000050     |
| AS\$CHK= 000000  | CE.RTE= 100376   | CS.RSN= 010000    | D\$DELF 000045  | ESNLLA 000012    |
| AS\$CPS= 000000  | CE.SRC= 100364   | CS.SHU= 000001    | D\$DELV 000046  | ESNLNK 000000    |
| AS\$PRI= 000000  | CE.STP= 100352   | CS.SID= 000002    | D\$END = 000104 | ESNML 000040     |
| AS\$TRP= 000000  | CE.TME= 100354   | CS.STR= 000004    | D\$FNB 000034   | ESNMR 000024     |
| A.ACC 000020     | CE.TMO= 100374   | CS.SUC= 000001    | D\$HIOR 000024  | ESNMS 000030     |
| A.LNK 000000     | CE.UNS= 100344   | CS.TMO= 020000    | D\$HOST 000022  | ESNNOD 000002    |
| A.NAM 000002     | CF.CHN= 000001   | CS.XUR= 000004    | D\$INAC 000044  | ESNRT 000042     |
| A.PUD 000010     | CF.EOM= 000004   | CTRSSES= ***** GX | D\$INCT 000042  | ESNRTP 000005    |
| A.REM 000012     | CF.HDR= 000020   | CVTADD 000042R    | D\$IPL 000051   | ESNSEG 000010    |
| A.UCB 000010     | CF.LB = 100000   | CVTARE 000054R    | D\$LID 000020   | ESNTIM 000046    |
| BLDDSC 000532R   | CF.LLN= 000002   | CVSM\$K= 000003   | D\$LNAM 000006  | ESNUSE 000004    |
| RDLIMG 000610R   | CF.SOM= 000010   | CV\$31 = 000001   | D\$LNAM 000014  | ESSTRT 000006    |
| BLDSES 000412R   | CF.SYN= 000040   | CV\$32 = 000000   | D\$LST 000047   | ES\$PR= 000000   |
| CB.ACN 000114    | CF.TRN= 000100   | CV\$40 = 000002   | D\$MAXC 000064  | FC.CCP= 000020   |
| CB.ACT 000112    | CHKACK= ***** GX | CX.GDQ= 000001    | D\$MAXH 000066  | FC.CTL= 000000   |
| CB.CCB= 000002   | CHKADD 000000R   | CX.REM= 000020    | D\$MAXV 000070  | FC.KCP= 000016   |
| CB.DDM= 000040   | CHKNAM 001256R   | CX.REQ= 000002    | D\$MML 000070   | FC.KIL= 000004   |
| CB.DFM 000006    | CL\$MFL= 000010  | CX.RUI= 000040    | D\$MNOD 000041  | FC.MAN= 000024   |
| CB.DGR 000010    | CL\$SFL= 000004  | CX.SMC= 000010    | D\$NA 000062    | FC.MLD= 000026   |
| CB.DLC= 000020   | CL\$TYP= 000001  | CX.UNL= 000004    | D\$NBEA 000056  | FC.PCT= 000030   |
| CB.DL1 000010    | CL.MU1= 000001   | C\$SORE= 000400   | D\$NBRA 000054  | FC.PWR= 000022   |
| CB.DL2 000014    | CL.MU2= 000002   | C\$SRSH= 177564   | D\$NEND= 000054 | FC.RCE= 000002   |
| CB.DOB 000007    | CL.RES= 177774   | C.ADD 000034      | D\$NLN 000030   | FC.RCP= 000014   |
| CB.DR1 000012    | CMPINT 001414RG  | C.BID 000003      | D\$NN 000060    | FC.TIM= 000010   |
| CB.DR2 000016    | CM.CIR= 000002   | C.BUF 000014      | D\$OUTT 000043  | FC.XCP= 000012   |
| CB.DUS 000012    | CM.CON= 000200   | C.BUF1 000014     | D\$RETF 000050  | FC.XME= 000000   |
| CB.LGT 000156    | CM.FMT= 100000   | C.BUF2 000024     | D\$RNN 000002   | FLSRES= ***** GX |
| CB.NOD 000000    | CM.HRD= 000002   | C.CNT 000020      | D\$RTMR 000076  | FS.AST= 000000   |
| CB.OPD 000134    | CM.LIN= 000000   | C.CNT1 000020     | D\$SEG 000036   | FS.CIB= 002000   |
| CB.OPT 000136    | CM.LOO= 000001   | C.CNT2 000030     | D\$SER 000032   | FS.CRA= 001000   |
| CB.PSL 000100    | CM.XLO= 000004   | C.FLG 000022      | D\$SGL 000052   | FS.DIS= 013000   |
| CB.PSW 000102    | CP.DCF= 000040   | C.FLG1 000022     | D\$BUG= 177514  | FS.DVC= 001000   |
| CB.RDB= 000004   | CP.HDL= 000007   | C.FLG2 000032     | D\$ISK= 000000  | FS.ENB= 012000   |
| CB.RDD 000056    | CP.PS = 177400   | C.FNC 000010      | D\$L11= 000001  | FS.EXI= 001000   |
| CB.RQ1 000060    | CP.PSI= 000200   | C.LIN 000006      | D\$SYNC= 000000 | FS.GET= 006000   |
| CB.SDB= 000010   | CP.XCF= 000100   | C.LNK 000000      | D\$SYNM= 000000 | FS.HLT= 000000   |
| CB.SFM 000032    | CP.2FR= 000030   | C.MOD 000011      | ER\$ABM= 000010 | FS.INI= 000000   |
| CB.SGR 000034    | CS.ABO= 000100   | C.NSP 000004      | ER\$ABQ= 000046 | FS.KIL= 000000   |
| CB.SLI= 000100   | CS.BRO= 000002   | C.PRO 000042      | ER\$ABT= 000011 | FS.LCL= 100000   |
| CB.SL1 000034    | CS.BUF= 000200   | C.RSV 000002      | ER\$ACC= 000042 | FS.LTM= 001000   |
| CB.SL2 000040    | CS.CES= 000002   | C.STA 000007      | ER\$CDI= 000052 | FS.MNT= 004000   |
| CB.SOB 000033    | CS.CHN= 000010   | C.STS 000012      | ER\$COM= 000047 | FS.MSN= 014000   |
| CB.SR1 000036    | CS.CMP= 000200   | C.URM 177776      | ER\$FMT= 000005 | FS.REA= 001000   |
| CB.SR2 000042    | CS.DCR= 000400   | C.XACP 000004     | ER\$MLB= 000006 | FS.RET= 000000   |
| CB.SUS 000036    | CS.DEF= 000004   | C.XID 000035      | ER\$NNF= 000012 | FS.REZ= 003000   |
| CB.XLB= 000001   | CS.DEV= 000002   | C.XLEN 000044     | ER\$NOD= 000002 | FS.RLB= 002000   |
| CC.LLC= 000200   | CS.DIS= 000040   | C.XPLI 000040     | ER\$NSL= 000013 | FS.RNG= 011000   |
| CEACC = ***** GX | CS.ENA= 000001   | C.XPT 000034      | ER\$NSR= 000003 | FS.RST= 000000   |
| CEMUL = ***** GX | CS.ENB= 000020   | C.XSVC 000042     | ER\$RES= 000001 | FS.RTN= 001000   |
| CE.ABO= 100362   | CS.ERR= 100000   | C.XTC 000037      | ER\$STA= 000051 | FS.SET= 005000   |
| CE.DAO= 100346   | CS.FTL= 001000   | C.X25 000036      | ER\$UOB= 000004 | FS.SFC= 005000   |
| CE.DIS= 100366   | CS.HCR= 000001   | DECP = ***** GX   | ENBR 000014     | FS.SFR= 006000   |

DLXCEX - DLX/CEX interface rout MACRO V05.03b Friday 28-Jun-85 22:53<sup>1 3</sup> Page 6  
Define local macros

```
53 .SBTTL Define local macros
54
55 .MACRO $DQPKT,QUEUE
56 JSR RO,DQPKT
57 .WORD L$QUEUE
58 .ENDM $DQPKT
```

DLXCEX - DLX/CEX interface rout MACRO V05.03b Friday 28-Jun-85 22:53<sup>J 3</sup> Page 7  
Process dispatch table

```

461                                     .SBTTL Characteristics complete processing
462                                     ;+
463                                     **~CHARC~Characteristics complete processing
464                                     :
465                                     This routine is called when the DLC has completed processing of a
466                                     set or get characteristics function.
467                                     :-
468                                     Inputs:
469                                     R1 = Completion status
470                                     R2 = Error offset for characteristics functions
471                                     R5 = Address of physical link access block
472                                     :
473 001140 010100 CHAR: MOV      R1,R0          ; Set up register conventions
474 001142 010201      MOV      R2,R1          ;      for I/O completion
475 001144      $DQPKT  FNC          ; Dequeue a control function
476 001152      BCS      10$          ; If CS, none available!
477 001154      CALLR   RQDON        ; Complete the request
478
479 001160      10$:   RETURN

```



```

1      .TITLE  DLXCHR - DLX Auxilliary characteristics processing
2      .IDENT  /V05.00/
3      .ENABL  LC
4
5      ; Copyright (C) 1982, 1983, 1985 by
6      ; Digital Equipment Corporation, Maynard, MASS.
7
8      ; This software is furnished under a license for use only on a
9      ; single computer system and may be copied only with the
10     ; inclusion of the above copyright notice. This software, or
11     ; any other copies thereof, may not be provided or otherwise
12     ; made available to any other person except for use on such
13     ; system and to one who agrees to these license terms. Title
14     ; to and ownership of the software shall at all times remain
15     ; in DEC.
16
17     ; The information in this document is subject to change without
18     ; notice and should not be construed as a commitment by Digital
19     ; Equipment Corporation.
20
21     ; DEC assumes no responsibility for the use or reliability of
22     ; its software on equipment which is not supplied by DEC.
23
24     ; Module description
25
26     ;       DLX auxilliary characteristics processing
27
28     ; Distributed Systems Software Engineering
29
30     ; Ident History:
31
32     4.00  07-NOV-83
33           DECNET-11M V4.0
34           DECNET-11M-PLUS V2.0
35
36     5.00  22-JUL-85
37           DECnet-11M/S V4.2
38           DECnet-11M-Plus V3.0
39           DECnet-Micro/RSX V1.0
40
41
42
43
44
45
46     .MCALL  SAVRG,RESRG,MAP
47     .MCALL  CCBDF$,CHRDf$,PLADf$
48
49     CCBDF$      ; Define CCB offsets
50     CHRDf$      ; Define characteristics symbols
51     PLADf$      ; Define PLA offsets

```

|     |     |   |
|-----|-----|---|
| 5-  | 44  | Macro definitions                             |
| 6-  | 54  | Define local macros                           |
| 7-  | 61  | LLC to LLC control functions                  |
| 8-  | 80  | Dispatch tables                               |
| 9-  | 111 | Control QIO function processing               |
| 10- | 151 | Turn <Line-id> on                             |
| 11- | 178 | Reassign <Line-id> for service access         |
| 12- | 246 | Common error routines                         |
| 13- | 283 | Assign <Line-id> to another LLC               |
| 14- | 320 | Turn <Line-id> off                            |
| 15- | 345 | Deassign <Line-id> from an LLC                |
| 16- | 373 | Set owner for <Line-id>                       |
| 17- | 401 | Set network management state for <Line-id>    |
| 17- | 421 | Set network management substate for <Line-id> |
| 18- | 449 | Queue a control CCB to the target LLC         |
| 19- | 474 | Control complete processing                   |
| 20- | 520 | Continue with reassignment                    |
| 21- | 552 | Continue with deassign operation              |
| 22- | 570 | Continue with off processing                  |
| 23- | 591 | Complete the control request                  |

```

345                                     .SBTTL Deassign <Line-id> from an LLC
346
347                                     ;+
348                                     **--CTLDEA=Deassign <Line-id> from an LLC
349                                     :
350                                     This routine is called to deassign the <Line-id> from it's owning
351                                     LLC.
352                                     -
353                                     Inputs:
354                                     R1 = Address of system line table
355                                     R3 = Address of I/O packet
356
357                                     Registers modified:
358                                     R0, R1, R2, R3, R4, R5
359 000532 005767 000016G CTLDEA: TST    $PBLK+T$ASN    ; Check line assignment
360 000536 001677          BEQ    ISALT        ; If EQ, line not assigned
361
362          000540          CALL    @CCBG7      ; Allocate a CCB
363 000544 103713          BCS    IENOD        ; If CS, allocation failure
364
365 000546 112764 000006 000004        MOV8    #NX.DEA,C.NSP(R4)
366
367 000554 016702 000010G CTLOF1: MOV    $PBLK+T$SLN,R2    ; Get SLN and logical tributary
368 000560 016764 000016G        MOV    $PBLK+T$ASN,C.LIN(R4)
369 000566 012764 002006 000010        MOV    #FC.CTL!FS.STP,C.FNC(R4)
370 000574 005064 000012          CLR    C.STS(R4)    ; Indicate stop and disable required
371 000600          CALLR   QCCBL        ; Queue the CCB to the target LLC

```

DLXCTL      CREATED BY MACRO ON 28-JUN-85 AT 22:54      PAGE 3      I 8

SYMBOL CROSS REFERENCE      CREF 04.00

| SYMBOL  | VALUE    | REFERENCES   |
|---------|----------|--|
| L.FLG   | 000000   | #5-52  |
| L.KRBA  | 000016   | #5-52  |
| L.LEN   | = 000022 | #5-52  |
| L.MPF   | 000022   | #5-52  |
| L.NMST  | 000020   | #5-52  |
| L.NSTA  | 000014   | #5-52  |
| L.OWNR  | 000021   | #5-52  |
| L.UNT   | 000013   | #5-52  |
| MAPAD   | = *****  | 20-539   |
| NXCTL   | 000044   | #9-127   |
| NX.ASS  | = 000000 | G 11-240 #19-509   |
| NX.DEA  | = 000006 | G 15-365 #19-512   |
| NX.DON  | = 000010 | G 13-315 #19-513   |
| NX.OFF  | = 000004 | G 14-342 #19-511   |
| NX.REA  | = 000002 | G #19-510 20-542   |
| N\$BUF  | = 000001 | 9-138  |
| OFF     | = 000000 | G #7-66  |
| PRCLD   | = *****  | GX 10-168 14-337   |
| QCCBL   | 000710   | RG 11-244 13-318 15-371 #18-467 20-547                         |
| RQALT   | = *****  | GX 12-279  |
| RQDON   | = *****  | GX 23-616  |
| R\$MPL  | = *****  | 19-515   |
| R\$11D  | = *****  | 5-50   |
| R\$11M  | = 000000 | 5-50   |
| R\$11S  | = *****  | 5-50   |
| SFMAX   | = 000022 | #8-95 9-131  |
| SF.ACT  | = 000200 | #5-52  |
| SF.ENA  | = 000100 | #5-52  |
| SF.LPB  | = 000004 | #5-52  |
| SF.MFL  | = 000040 | #5-52  |
| SF.PAC  | = 000020 | #5-52  |
| SF.REA  | = 000010 | #5-52  |
| SF.SER  | = 000001 | #5-52  |
| SF.SVC  | = 000002 | #5-52  |
| SF.UNL  | = 000040 | #5-52  |
| SRCTBL  | 000000   | R #8-85 8-95 9-135   |
| STACT   | = *****  | GX 10-174  |
| STNMST  | = *****  | GX 16-397 17-439 22-586  |
| STOP    | = 000001 | G #7-65 11-243   |
| STRT    | = 000004 | G #7-71 10-175   |
| S.COST  | 000001   | #5-52  |
| S.FLG   | 000000   | #5-52  |
| S.LEN   | 000004   | #5-52  |
| S.NMST  | 000002   | #5-52  |
| S.OWNR  | 000003   | #5-52  |
| TSTBF   | = *****  | GX 9-141   |
| T\$ASN  | 000016   | 10-165 11-227 11-237 11-241 13-298 13-316 14-334 15-359 15-368 |
| T\$NMST | 000022   | 11-198   |
| T\$SLN  | 000010   | 10-173 11-236 13-309 15-367 16-396 17-438                      |
| X\$MCB  | = *****  | 5-50   |
| ZF.COJ  | = 001000 | #5-50  |
| ZF.DDM  | = 000001 | #5-50  |

DLXDAT      CREATED BY MACRO ON 28-JUN-85 AT 22:54      PAGE 3      1 9  
MACRO CROSS REFERENCE      CREF      04.00  
MACRO NAME      REFERENCES  
PLADF\$      #5-46      5-48

DLXLIN      CREATED BY    MACRO    ON 28-JUN-85 AT 22:55      PAGE 3      I 10  
SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE         | REFERENCES |
|--------|---------------|------------|
| ZF.DVP | = 100000      | #5-50      |
| ZF.INI | = 040000      | #5-50      |
| ZF.KMX | = 000020      | #5-50      |
| ZF.LLC | = 000004      | #5-50      |
| ZF.LMC | = 000100      | #5-50      |
| ZF.MAN | = 020000      | #5-50      |
| ZF.MFL | = 000010      | #5-50      |
| ZF.MTM | = 000400      | #5-50      |
| ZF.MUX | = 000040      | #5-50      |
| ZF.PSE | = 002000      | #5-50      |
| ZF.SLI | = 010000      | #5-50      |
| ZF.TIM | = 000200      | #5-50      |
| ZF.X3P | = 000000      | #5-50      |
| ZS.ASN | = 100000      | #5-50      |
| ZS.BSY | = 140000      | #5-50      |
| Z.AVL  | 000014        | #5-50      |
| Z.DAT  | 000016        | #5-50      |
| Z.DSP  | 000000        | #5-50      |
| Z.FLG  | 000010        | #5-50      |
| Z.LEN  | = 000016      | #5-50      |
| Z.LLN  | 000006        | #5-50      |
| Z.MAP  | 000020        | #5-50      |
| Z.NAM  | 000004        | #5-50      |
| Z.PCB  | 000012        | #5-50      |
| Z.SCH  | 000007        | #5-50      |
| BLLCRQ | = *****    GX | 6-76       |

5-50

DLXQIO - ISSUE I/O TO DLX  
DLXLUN - ASSIGN A LUN TO NX:

MACRO V05.03b Saturday 29-Jun-85 12:23 Page 9

J 11

186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200 000200  
201 000200  
202 000202  
203 000206  
204 000226 103415  
205 000230 062704 000166  
206 000234  
207 000250 103404  
208 000252 105764 000003  
209 000256 100401  
210 000260 000261  
211 000262  
212 000264  
213  
214 000001

```
.SBTTL DLXLUN - ASSIGN A LUN TO NX:
:
: **--DLXLUN-ASSIGN A LUN TO NX:
:
: THIS ROUTINE IS CALLED TO ASSIGN $TMLUN TO NX:.
:
: OUTPUTS:
: IF CC, NX: IS LOADED AND ALUN SUCCEEDED.
: ELSE, NX: NOT PRESENT
:
: REGISTERS:
: NO REGISTERS MODIFIED
:
:
DLXLUN::
    SAVRG <R4>          ; SAVE R4
    CALL  DEALUN         ; DEASSIGN THE TEMPORARY LUN
    ALUN$$ # $TMLUN, #'NX ; TRY TO ASSIGN A LUN TO NX:
    BCS   10$            ; IF CS, NOT THERE
    ADD   #L$SCR,R4      ; POINT TO SCRATCH BUFFER
    GLUN$$ # $TMLUN,R4   ; GET LUN INFORMATION
    BCS   10$            ; IF CS, ASSUME NO DLX
    TSTB  3(R4)          ; IS THE DRIVER LOADED ?
    BMI   10$            ; IF MI, YES
    SEC                     ; ELSE, NO
10$:    RESRG <R4>       ; RESTORE R4
    RETURN
:
: .END
```

DLXSUB - DLX subroutines  
Assign line to an LLC process

MACRO V05.03b Friday 28-Jun-85 22:56 Page 6-1

110  
111 000102  
112  
113

20\$: RETURN  
.DSABL LSB



Find free channel in process mapping table

```

641 .SBTTL Find free channel in process mapping table
642
643 *
644 **FNDFR-Find free channel in process mapping table
645
646 Scan the channel mapping table at the end of the PDV to find an
647 unused entry.
648
649 Inputs:
650 R2 = PDV index of the process
651
652 Outputs:
653 R0 = PDV and Channel
654 Error code on failure:
655 IE.RSU - Line is in loopback
656 IE.NOD - No free channels available
657 R1 = Address of PDV channel entry
658 'C' Clear - Free channel found
659 'C' Set - No free channels available or line is in loopback
660
661 FNDFR:: SAVRG <RS> ; Get a free register
662 MOV #IE.RSU&377,R0 ; Assume line is in loopback
663 BIT #LF.MTP,$PBLK+T$SLF
664 BEQ 10$ ; If EQ, not a multipoint line
665 TSTB $PBLK+T$NSTA ; Is this a multipoint master?
666 BNE 20$ ; If NE, yes
667
668 10$: BIT #LF.LPB,$PBLK+T$SLF
669 BR 30$ ; Enter common code
670 20$: BIT #SF.LPB,$PBLK+T$SFLG
671 BNE 70$ ; If NE, line or tributary is in loopback
672
673 MOV R2,R1 ; Compute address of entry in
674 ADD @PDVTA,R1 ; PDV index table
675 MOV (R1),R1 ; Get pointer to the PDV
676 ADD #Z.MAP,R1 ; Point to the channel mapping table
677 CLR R0 ; Initialise channel #
678 CLR R5 ; Get # of channels to scan
679 BISB Z.LLN-Z.MAP(R1),R5
680 BEQ 60$ ; If EQ, no channels for this process
681
682 40$: MOV (R1)+,-(SP) ; Get next entry
683 ASL (SP)+ ; Is it free to assign?
684 BCC 50$ ; If CC, no
685 BPL 80$ ; If PL, yes ... assign it
686
687 50$: INC R0 ; Update channel #
688 SOB R5,40$ ; Scan all channel entries
689
690 60$: MOV #IE.NOD&377,R0 ; Set up error code
691 SEC ; Indicate error
692 BR 90$ ; and exit
693
694 80$: SWAB R0 ; Merge PDV index and channel #
695 BIS R2,R0 ; ...
696 SWAB R0 ; ...
697 TST -(R1) ; Backup channel entry address (Clears C-bit)

```

```

1075                                     .SBTTL  Compute address of the system line table
1076                                     ;+
1077                                     ;**--STSLT-Compute address of the system line table
1078                                     ;
1079                                     ;   Compute the address of the system line table associated with
1080                                     ;   <Line-id>.
1081                                     ;-
1082                                     ; Inputs:
1083                                     ;   R2 = SLN and logical tributary #
1084                                     ;
1085                                     ; Outputs:
1086                                     ;   R5 = Address of the system line table
1087                                     ;
1088 002602 010205 STSLT:: MOV     R2,R5      ; Isolate SLN
1089 002604 042705      BIC     #<377>,R5    ; ...
1090 002610 006305      ASL     R5            ; Form word offset
1091 002612 067705      ADD     @SLTMA,R5     ; Point into system line index table
1092 002616 011505      MOV     (R5),R5      ; Get address of system line table entry
1093 002620
1094
1095      000001      .END

```

EVL - EVENT LOGGER PROCESS  
DEFINE LOCAL MACROS

MACRO V05.03b Friday 28-Jun-85 22:58 Page 6

J 15

65  
66  
67  
68  
69  
70  
71  
72  
73

.SBTTL DEFINE LOCAL MACROS

.MACRO EVENT,CLASS,TYPE  
.WORD <CLASS\*100>+TYPE  
.ENDM EVENT

.MACRO ENDEVENT  
.WORD 0  
.ENDM ENDEVENT

|                  |                 |                 |                  |                 |
|------------------|-----------------|-----------------|------------------|-----------------|
| AC\$DNT= 000002  | CL\$XL3= 013600 | C.BUF2 000024   | EV\$INF= 000515  | E.LCN 000042    |
| AC\$X25= 000001  | CL\$X2S= 013500 | C.CNT 000020    | EV\$LDL= 000407  | E.LEN 000216    |
| AE\$CIR= 000003  | CM.CIR= 000002  | C.CNT1 000020   | EV\$LDN= 010416  | E.LIN 000024    |
| AE\$LIN= 000001  | CM.FMT= 100000  | C.CNT2 000030   | EV\$LDO= 000411  | E.LNK 000000    |
| AE\$MOD= 000004  | CM.HRD= 000002  | C.FLG 000022    | EV\$LDS= 000410  | E.MOD 000036    |
| ALLOD 000532R    | CM.LIN= 000000  | C.FLG1 000022   | EV\$LSC= 000500  | E.NOD 000034    |
| AS\$CHK= 000000  | CM.LOO= 000001  | C.FLG2 000032   | EV\$SLUP= 000412 | E.PDV 000021    |
| AS\$CPS= 000000  | CM.XLO= 000004  | C.FNC 000010    | EV\$NOL= 000402  | E.PORT 000040   |
| AS\$PRI= 000000  | CP.DCF= 000040  | C.LIN 000006    | EV\$NRC= 000416  | E.PRM 000026    |
| AS\$TRP= 000000  | CP.HDL= 000007  | C.LNK 000000    | EV\$NSC= 000200  | E.PVC 000044    |
| CB.CCB= 000002   | CP.PS = 177400  | C.MOD 000011    | EV\$NUL= 000401  | E.SIZ 000022    |
| CB.DDM= 000040   | CP.PSI= 000200  | C.NSP 000004    | EV\$NVR= 000406  | E.TIME 000004   |
| CB.DLC= 000020   | CP.XCF= 000100  | C.PRO 000042    | EV\$OPL= 000403  | FC.CCP= 000020  |
| CB.RDB= 000004   | CP.2FR= 000030  | C.RSV 000002    | EV\$PCC= 034000  | FC.CTL= 000006  |
| CB.SDB= 000010   | CS.ABO= 000100  | C.STA 000007    | EV\$PCI= 034001  | FC.KCP= 000016  |
| CB.SLI= 000100   | CS.BRO= 000002  | C.STS 000012    | EV\$PCM= 034002  | FC.KIL= 000004  |
| CB.XLB= 000001   | CS.BUF= 000200  | C.URM 177776    | EV\$PFE= 000404  | FC.MAN= 000024  |
| CC.LLC= 000200   | CS.CES= 000002  | C.XACP 000004   | EV\$PPC= 034003  | FC.MLD= 000026  |
| CEACC 000014R    | CS.CHM= 000010  | C.XID 000035    | EV\$RCF= 000517  | FC.PCT= 000030  |
| CE.ABO= 100362   | CS.CMP= 000200  | C.XLEN 000044   | EV\$RDC= 010001  | FC.PWR= 000022  |
| CE.DAO= 100346   | CS.DCR= 000400  | C.XPLI 000040   | EV\$RDR= 010002  | FC.RCE= 000002  |
| CE.DIS= 100366   | CS.DEF= 000004  | C.XPT 000034    | EV\$RJE= 035106  | FC.RCP= 000014  |
| CE.ERR= 100370   | CS.DEV= 000002  | C.XSVC 000042   | EV\$RSC= 000501  | FC.TJM= 000010  |
| CE.ILN= 100350   | CS.DIS= 000040  | C.XTC 000037    | EV\$RUL= 000405  | FC.XCP= 000012  |
| CE.LTO= 100356   | CS.ENA= 000001  | C.X25 000036    | EV\$SNA= 035000  | FC.XME= 000000  |
| CE.MOP= 100372   | CS.ENB= 000020  | DL\$AST= 000002 | EV\$SNF= 000516  | FR\$BCC= 000007 |
| CE.NTE= 100361   | CS.ERR= 100000  | DL\$HLT= 000000 | EV\$SPE= 035001  | FR\$CCF= 000001 |
| CE.RTE= 100376   | CS.FTL= 0010J3  | DL\$IST= 000001 | EV\$XCE= 034110  | FR\$CDF= 000002 |
| CE.SRC= 100364   | CS.HCR= 000001  | DL\$MAI= 000004 | EV\$XDI= 013600  | FR\$DAO= 000011 |
| CE.STP= 100352   | CS.HFE= 002000  | DL\$OFF= 000001 | EV\$XGW= 034111  | FR\$EXC= 000000 |
| CE.TME= 100354   | CS.LST= 040000  | DL\$ON= 000000  | EV\$XMX= 000514  | FR\$FRM= 000010 |
| CE.TMO= 100374   | CS.MTL= 004000  | DL\$RUN= 000003 | EV\$XRS= 000512  | FR\$FTL= 000005 |
| CE.UNS= 100344   | CS.RNG= 000010  | DL\$SHU= 000002 | EV\$XSC= 000513  | FR\$OPN= 000004 |
| CF\$BLK= 000102  | CS.ROV= 000004  | DL\$SYN= 000005 | EV\$X2S= 013500  | FR\$RFD= 000006 |
| CF.CHN= 000001   | CS.RSN= 010000  | D\$BUG= 177514  | EV.CCB= 000001   | FR\$SBU= 000012 |
| CF.EOM= 000004   | CS.SHU= 000001  | D\$ISK= 000000  | EV.CIR= 000020   | FR\$SHO= 000003 |
| CF.HDR= 000020   | CS.SID= 000002  | D\$SL11= 000001 | EV.LCB= 000100   | FR\$UBU= 000013 |
| CF.LB = 100000   | CS.STR= 000004  | D\$SYNC= 000000 | EV.LIN= 000004   | FR\$UPT= 000014 |
| CF.LIN= 000002   | CS.SUC= 000001  | D\$SYNM= 000000 | EV.MAP= 000002   | FS.AST= 000000  |
| CF.SOM= 000010   | CS.TMO= 020000  | EF\$ACT= 000001 | EV.MOD= 000040   | FS.CIB= 002000  |
| CF.SYN= 000040   | CS.XUR= 000004  | EV\$ACF= 000201 | EV.NOD= 000010   | FS.CRA= 001000  |
| CF.TRN= 000100   | C\$DTL= 000012  | EV\$ADR= 000420 | EV.PRT= 000200   | FS.DIS= 013000  |
| CL\$ASZ= 010500  | C\$FLG 000015   | EV\$ADU= 000417 | EXRON 000016R    | FS.DVC= 001000  |
| CL\$DLL= 000500  | C\$GNAM 000016  | EV\$APL= 000400 | \$DATA 000020    | FS.ENB= 012000  |
| CL\$ECL= 000300  | C\$LCN 000010   | EV\$ARC= 000421 | \$EVTs 000000    | FS.EXI= 001000  |
| CL\$LDN= 010400  | C\$LEN 000016   | EV\$AUC= 000010 | \$EVLN 000016    | FS.GET= 006000  |
| CL\$MAN= 000000  | C\$LNK 000000   | EV\$AUS= 000003 | \$SLIN 000000    | FS.HLT= 000000  |
| CL\$PAZ= 034100  | C\$NAM 000002   | EV\$CDF= 000520 | \$SMOD 000012    | FS.INI= 000000  |
| CL\$PLH= 034000  | C\$NML 000002   | EV\$COZ= 000011 | \$NOD 000010     | FS.KIL= 000000  |
| CL\$PLL= 000600  | C\$PORT 000014  | EV\$DBR= 000302 | \$PORT 000014    | FS.LCL= 100000  |
| CL\$PRT= 034200  | C\$CKP= 000000  | EV\$GAS= 035101 | \$PRM 000002     | FS.LTM= 001000  |
| CL\$ROU= 010000  | C\$ORE= 000400  | EV\$HCE= 035114 | \$STAT 000006    | FS.MNT= 004000  |
| CL\$SES= 000200  | C\$SRSH= 177564 | EV\$HCI= 035113 | \$TCB 000004     | FS.MSN= 014000  |
| CL\$SGE= 035000  | C.ADD 000034    | EV\$HFE= 000506 | \$XPR= 000000    | FS.REA= 001000  |
| CL\$SSE= 035100  | C.BID 000003    | EV\$IFL= 000413 | E.CTL 000020     | FS.RET= 000000  |
| CL\$STRN= 000400 | C.BUF 000014    | EV\$IFO= 000415 | E.DATA 000046    | FS.REZ= 003000  |
| CL\$XL2= 013700  | C.BUF1 000014   | EV\$IFS= 000414 | E.EVT 000002     | FS.RLB= 002000  |

```

287 .SBTTL Build process/task descriptor
288
289 **-BLDDSC-Build process/task descriptor
290
291 Build a descriptor for the source/destination process/task in the
292 message buffer.
293
294 Inputs:
295 R0 = Address of descriptor
296 R2 = Pointer to next available byte in message buffer
297
298 Outputs:
299 R2 = Pointer past last byte used in message buffer
300 'C' Clear - Valid descriptor built
301 'C' Set - Format error
302
303 Registers modified:
304 R0, R1, R5
305
306 000532 012701 000020 BLDDSC: MOV #16,,R1 ; Set max length of field
307 000536 112005 MOV (R0)+,R5 ; Get the format type code
308 000540 110522 MOV R5,(R2)+ ; and plant in the buffer
309 000542 112022 MOV (R0)+,(R2)+ ; Copy object type
310
311 000544 022705 CMP #NC.FM2,R5 ; Is the format type valid?
312 000550 103416 BLO 20$ ; If L0, no (C-bit set)
313 000552 005305 DEC R5 ; Check for format type
314 000554 100414 BMI 20$ ; If M1, type 0 - all done
315 000556 001406 BEQ 10$ ; If EQ, type 1 - just copy image data
316
317 000560 112022 MOV (R0)+,(R2)+ ; Copy group code
318 000562 112022 MOV (R0)+,(R2)+ ; ...
319 000564 112022 MOV (R0)+,(R2)+ ; Copy i code
320 000566 112022 MOV (R0)+,(R2)+ ; ...
321 000570 012701 000014 MOV #12,,R1 ; Set new max length of field
322
323 000574 020110 10$: CMP R1,(R0) ; Is the field too long?
324 000576 103403 BLO 20$ ; If L0, yes (C-bit set)
325 000600 012001 MOV (R0)+,R1 ; Get length of field to copy
326 000602 MOVIMG ; Copy the field
327
328 000606 20$: RETURN

```

|                    |                 |                 |                 |                  |
|--------------------|-----------------|-----------------|-----------------|------------------|
| FS.SFS= 004000     | LT.DIR= 000010  | MA.CI = 000040  | NE\$TCD= 000006 | N\$FLG 000005    |
| FS.SPW= 040000     | LT.LCL= 000001  | MA.DA = 000000  | NE\$TPA= 000010 | N\$FNC 000006    |
| FS.STM= 000000     | LT.LPL= 000002  | MA.IL = 000020  | NE\$UOB= 000004 | N\$GENO 000052   |
| FS.STP= 002000     | LT.NOT= 000040  | MC.CC = 000040  | NF\$BLK= 000100 | N\$GTM 000015    |
| FS.STR= 001000     | LT.RSU= 000200  | MC.CI = 000020  | NF\$DMO= 000010 | N\$HIGH 000033   |
| FS.TRM= 003000     | LT.SLI= 000004  | MC.DC = 000100  | NF\$MOU= 000040 | N\$LLT 000026    |
| FS.WLB= 001000     | LT.TDA= 000100  | MC.DI = 000060  | NF\$RST= 000002 | N\$LLTM 000024   |
| FS.XKL= 002000     | L\$ASG= 000000  | MC.NO = 000000  | NF\$SCN= 000020 | N\$LVC 000036    |
| FS.XOF= 010000     | L\$DRV= 000000  | MC.RC = 000140  | NF\$SHU= 000004 | N\$MBXO 000050   |
| FS.XON= 007000     | L\$P11= 000001  | MD.BM = 000040  | NF\$TIM= 000200 | N\$PLLT 000030   |
| FS.ZER= 002000     | L\$11R= 000000  | MD.EM = 000100  | NM\$ARA= 176000 | N\$SLA 000016    |
| F\$1VL= 000001     | L.CSTA 000037   | MD.ILS= 000040  | NM\$NOD= 001777 | N\$SNDD 000012   |
| GETLDB= ***** GX   | L.CTR 000074    | MD.IM = 000020  | ND.DTR= 000077  | N\$TIM 000004    |
| GETSDB= ***** GX   | L.DCR 000100    | MF.ACK= 000004  | NO.FAL= 000021  | N\$VCB 000010    |
| G\$1TP= 000000     | L.FLAG 000014   | MF.CTL= 000010  | NO.FA1= 000001  | N\$1ACC= 000001  |
| G\$1TSS= 000000    | L.ILSO 000052   | MF.DAT= 000000  | NO.NCU= 000023  | N\$1ACK= 000011  |
| G\$1TTK= 000000    | L.ILTT 000066   | MOVING 00624R   | ND.RTL= 000022  | N\$1EVL= 000001  |
| G\$1WRD= 000000    | L.LDA 000032    | M\$1CRB= 000124 | NO.TAS= 000000  | N\$1HDR= 000007  |
| HOST = ***** GX    | L.LIA 000034    | M\$1CRX= 000000 | NO.TCL= 000017  | N\$1LDV= 000001  |
| IN.DAT= 000400     | L.LLA 000002    | M\$1FCS= 000000 | NO.TC1= 000005  | N\$1MLL= 000001  |
| IN.ILS= 000001     | L.LNG 000124    | M\$1MGE= 000000 | NO.TLK= 000020  | N\$1MOV= 000010  |
| I\$1RAR= 000000    | L.LNO 000026    | M\$1MUP= 000000 | N\$1DON= 000000 | N\$1NCT= 000001  |
| I\$1RDN= 000000    | L.LPT 000065    | M\$1NET= 000000 | N\$1SDI= 000002 | N\$1OVR= 000022  |
| I.TCB = ***** GX   | L.LSA 000030    | M\$1OVR= 000000 | N\$1WDC= 000004 | N\$1PEM= 000001  |
| K1SAR6= ***** GX   | L.LSFD 000046   | M.MAIL 000014   | NT\$AKD= 000020 | N\$1SES= 000001  |
| K\$1CNT= 177546    | L.LSFI 000044   | M.MAX 000011    | NT\$AKI= 000022 | N.RAC 000070     |
| K\$1CSR= 177546    | L.LTT 000062    | M.MBL 000020    | NT\$CC = 000016 | N.RACC 000066    |
| K\$1LDC= 000000    | L.MASQ 000070   | M.NAST 000007   | NT\$CON= 000000 | N.RDE 000012     |
| K\$1TPS= 000074    | L.MAST 000073   | M.NEXT 000002   | NT\$CTL= 000000 | N.RDEC 000010    |
| LA.ACK= 100000     | L.MASZ 000072   | M.RESP 000016   | NT\$DAT= 000002 | N.RFM 000006     |
| LA.CRS= 020000     | L.NIN 000020    | M.SPA 000012    | NT\$DC = 000012 | N.RGP 000010     |
| LA.MSK= 170000     | L.NXN 000016    | M.TASK 000004   | NT\$DIS= 000014 | N.RID 000034     |
| LA.NAK= 110000     | L.NXTH 000010   | M.USE 000010    | NT\$DLS= 000006 | N.RIDC 000032    |
| LA.NMS= 010000     | L.OPD 000103    | NC.FMO= 000000  | NT\$ILS= 000010 | N.RND 000000     |
| LA.RES= 040000     | L.OPDL 000102   | NC.FM1= 000001  | NT\$IMS= 000022 | N.RNM 000016     |
| LA.WND= 004000     | L.REM 000006    | NC.FM2= 000002  | NT\$INT= 000004 | N.RNMC 000014    |
| LD\$BRT = ***** GX | L.RFC 000050    | NE\$ABM= 000010 | NT\$RET= 000032 | N.ROT 000007     |
| LD\$LP = 000000    | L.RLA 000004    | NE\$ABO= 000046 | NT\$ROU= 000024 | N.RPS 000056     |
| LF.DRD= 000004     | L.RNO 000022    | NE\$ACC= 000042 | NT\$RTR= 000030 | N.RPSC 000054    |
| LF.FRC= 000001     | L.RTQ 000060    | NE\$ACT= 000042 | NT\$TSP= 000026 | N.RQL = 000110   |
| LF.HFO= 000010     | L.RTYD 000055   | NE\$COM= 000047 | NT.ABD= 000005  | N.RUS 000012     |
| LF.HMF= 000040     | L.RTYI 000057   | NE\$FCF= 000050 | NT.ABT= 000004  | P\$1P45= 000000  |
| LF.HSF= 000020     | L.SEC 000064    | NE\$FMT= 000005 | NT.CON= 000001  | P\$1WRD= 000000  |
| LF.IRD= 000002     | L.SEGZ 000076   | NE\$GEN= 000007 | NT.DSC= 000003  | Q\$1OPT= 000010  |
| LF.MMF= 000200     | L.STA 000000    | NE\$IFC= 000030 | NT.EVT= 000006  | REMLNK= ***** GX |
| LF.MSF= 000100     | L.TC 000042     | NE\$ILS= 000043 | NT.INT= 000002  | RF.LOO= 100000   |
| LD\$CAL = ***** GX | L.TIC 000043    | NE\$IMG= 000042 | NT.MDP= 000010  | RTRSES= ***** GX |
| LS.DLS= 100000     | L.TIPD 000013   | NE\$MLB= 000006 | NT.NSP= 000010  | R\$1DER= 000000  |
| LS.FCC= 000004     | L.TIPI 000012   | NE\$NMF= 000012 | NT.VFY= 000007  | R\$1K11= 000001  |
| LS.FCO= 000001     | L.TMRD 000054   | NE\$NOD= 000002 | NSACQ 000000    | R\$1SSND= 000000 |
| LS.FCI= 000002     | L.TMRI 000056   | NE\$NSD= 000003 | NSACTL 000032   | R\$111M= 000000  |
| LS.ILS= 100000     | L.TYP 000001    | NE\$NSL= 000013 | NSCIR 000034    | R\$115= 000000   |
| LS.MAK= 000020     | L.USA 000024    | NE\$NSR= 000003 | NSDLA 000020    | R.ADD 000010     |
| LS.MNK= 000040     | L.USTA 000036   | NE\$RES= 000001 | NSDLY 000014    | R.FLAG 000012    |
| LS.RES= 000360     | L.VER 000015    | NE\$SSR= 000000 | N\$ELEN 000054  | R.LEN 000014     |
| LS.RSV= 000300     | L.WIND 000040   | NE\$SSS= 000045 | N\$ENC 000042   | R.LNK 000000     |
| LT.CCA= 000020     | MAPNAM 000642RG | NE\$TCN= 000040 | N\$ERRC 000022  | R.NAM 000002     |

```

60
61
62
63
64
65
66
67 000000 000001'
68 000002 000003'
69 000004 000005'
70 000006 000022'
71 000010 000232'
72 000012 000352'
73 000014 000444'
74 000016 000017'
75 000020 000706'

.SBTTL Process dispatch table
;+
;*-DLXTB-Process dispatch table
;
; This dispatch table provides the interface between the COMM/EXEC and
; the DLX process.
;-
$DLXTB::WORD .+1 ; Transmit enable (not supported)
           .WORD .+1 ; Receive enable (not supported)
           .WORD .+1 ; Kill enable (not supported)
           .WORD CTLEN ; Control enable
           .WORD TMOU ; Timeout
           .WORD XMTCP ; Transmit complete
           .WORD RCVCP ; Receive complete
           .WORD .+1 ; Kill complete (not supported)
           .WORD CTLCP ; Control complete

```

```

481                                     .SBTTL Stop complete processing
482
483     +
484     **--STOPC-Stop complete processing
485
486     This routine is called when a stop requested is completed by the DLC
487     layer.
488
489     -
490     Inputs:
491     R1 = Completion status
492     R2 = Error offset for characteristics functions
493     R5 = Address of physical link access block
494
495 STOPC:: CALL    XMTAB      ; Abort any pending transmits
496           TST     L$RSR(R5) ; Are all functions and transmits complete?
497           BNE     100$     ; If NE, no
498
499           MOVB    (R5),R0   ; Get current physical link state
500           MOVB    #L$IDL,(R5) ; and reset it to idle
501           BITB    #L$RST,R0 ; Should we restart the link?
502           BNE     80$       ; If NE, yes
503
504           MOV     L$FNC(R5),R3 ; Get pending control I/O packet
505           BEQ     20$       ; If EQ, none
506           CMPB    #7,I.FCN+1(R3) ; Is this a close LUN request?
507           BNE     30$       ; If NE, no
508           MOV     I.UCB(R3),R2 ; Decrement the volume use count
509           DEC     @U.VCB(R2) ; ...
510
511           .IF DF R$SMPL
512           MOV     I.TCB(R3),R2 ; Get task's TCB address
513           MOV     T.PCB(R2),R2 ; then PCB address for task's window 0
514           TST     P.HDR(R2) ; Is the header in primary pool?
515           BNE     10$       ; If NE, yes
516           MAP     P.REL(R2) ; Map to the task header
517
518           10$:
519           .ENDC
520
521           CLR     @I.LN2(R3) ; Deaccess the link
522           CLR     L$TCB(R5) ; Mark the link inactive
523           BR      30$       ; and enter common code
524
525           20$: CLR     L$TCB(R5) ; Mark the link inactive
526           BR      70$       ; Enter common code
527
528           30$: BIT     #LF$BRO,(R5) ; Broadcast channels should always
529           BNE     40$       ; be deassigned
530           BITB    #LR$DEA,I.FCN(R3) ; If PL, no deassign - leave link active
531           BPL     60$       ; If NE, no deassign - set link inactive
532           BNE     50$       ; Deassign the physical link
533           CALL    DSLIN
534
535           40$: CLR     L$TCB(R5) ; Mark link access block free
536           CALL    CTLDN      ; Complete the I/O request
537
538           50$: CLR     L$TCB(R5) ; Mark link access block free
539           CALL    CTLDN      ; Complete the I/O request
540
541           60$: CALL    CTLDN      ; Complete the I/O request
542
543           70$: MOV     L$(TL(R5),R4 ; Is there a pending control CCB?
544           BEQ     80$       ; If EQ, no
545
546           80$:

```



```

52      .SBTTL Process receive auxilliary characteristics
53      ;+
54      **--RCVCHR-Process receive auxilliary characteristics
55      ;
56      Process the characteristics present in the auxilliary characteristics
57      buffer on the receive request.
58      ;
59      --
60      Inputs:
61      R3 = Address of the I/O packet
62      I.PRM+6 - Bias of the characteristics buffer
63      I.PRM+10- Virtual address of the characteristics buffer
64      I.PRM+12- Length of the characteristics buffer
65      I.PRM+14- Virtual address of byte past end of characteristics buffer
66      The characteristics buffer is mapped.
67      R4 = Address of the receive CCB
68      C.BUF - Bias of received message
69      C.BUF+2 - Virtual address of received message
70      C.CNT - Length of received message
71      C.NSP - Virtual address of Ethernet header
72      C.PRO - Protocol type from message
73      C.ADD - Source address from Ethernet message
74      R5 = Address of the physical link access block
75      ;
76      Registers modified:
77      R0, R1, R2
78      RCVCHR::MOV I.PRM+10(R3),R0 ; Point to start of characteristics buffer
79      ;
80      000000 016300 000010G
81      80 000004 012701 000002 10$: MOV #CS.IGN,R1 ; Assume characteristic will be ignored
82      81 000010 011002 ; MOV (R0),R2 ; Get next characteristics type code
83      82 000012 162702 000100 SUB #CC.ADR,R2 ; Normalise for DLX functions
84      83 000016 002410 BLT 20$ ; If LT, not one of our characteristics
85      84 000020 006302 ASL R2 ; Form word offset
86      85 000022 020227 000006 CMP R2,#RCMAX ; In range?
87      86 000026 103004 BHS 20$ ; If HS, no
88      87 000030 005060 000004 CLR C.DATO(R0) ; Assume no data available
89      88 000034 CALL @RCVTBL(R2) ; Dispatch to processing routine
90      90 000040 010160 000006 20$: MOV R1,C.STAT(R0) ; Return status of this characteristic
91      91 000044 066000 000002 ADD C.DATI(R0),R0 ; Point to next characteristic
92      92 000050 062700 000010 ADD #C.CHRL,R0 ; ...
93      93 000054 020063 000014G CMP R0,I.PRM+14(R3) ; Are we done yet?
94      94 000060 103751 BLO 10$ ; If LO, no
95      95 000062 RETURN
96      ;
97      ;+
98      --
99      ; Receive characteristics dispatch table
100     000064 000072' RCVTBL: .WORD RCVADR ; Retrieve Ethernet address
101     000066 000146' .WORD RCVPRO ; Retrieve Ethernet protocol
102     000070 000206' .WORD RCVDAD ; Retrieve destination Ethernet address
103     000006 RCMAX =.-RCVTBL

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

.TITLE DLXCTL - DLX control Q10 function processing  
.IDENT /V05.00/  
.ENABL LC

Copyright (C) 1982, 1983, 1985 by  
Digital Equipment Corporation, Maynard, MASS.

This software is furnished under a license for use only on a  
single computer system and may be copied only with the  
inclusion of the above copyright notice. This software, or  
any other copies thereof, may not be provided or otherwise  
made available to any other person except for use on such  
system and to one who agrees to these license terms. Title  
to and ownership of the software shall at all times remain  
in DEC.

The information in this document is subject to change without  
notice and should not be construed as a commitment by Digital  
Equipment Corporation.

DEC assumes no responsibility for the use or reliability of  
its software on equipment which is not supplied by DEC.

#### Module description

DLX control Q10 function processing

Distributed Systems Software Engineering

#### Ident history:

- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

\*

```

373                                     .SBTTL Set owner for <Line-id>
374
375     + **-CTLOWN-Set owner for <Line-id>
376
377     Set up the owner field of the network management word for the
378     <Line-id> to the specified process.
379
380     Inputs:
381         R1 = Address of system line table
382         R3 = Address of I/O packet
383             1.PRM+6 - RAD50 name of the owner LLC
384
385     Registers modified:
386         R0, R1, R2, R3, R4, R5
387
388     000604 016302 000006G   CTLOWN: MOV     1.PRM+6(R3),R2 ; Get the target process name
389     000610 001405          BEQ     10$ ; Zero means none
390
391     000612          CALL     FNDPR ; Find the process PDV index
392     000616 012700 000000C   MOV     #IE.MOD&377,R0 ; Assume it does not exist in the system
393     000622 103666          BCS     CTLERR ; If CS, report the error
394
395     000624 110246          10$: MOVB   R2,-(SP) ; Save PDV index for later
396     000626 016702 000010G   MOV     $PBLK+1$SLN,R2 ; Get SLN and logical tributary #
397     000632          CALL     STNMST ; Compute address of network management byte
398     000636 112665 000001   MOVB   (SP)+,1(R5) ; Set up line owner
399     000642          CALLR   ISSUC ; Complete the request successfully

```

## SYMBOL CROSS REFERENCE

CREF    04.00

| SYMBOL  | VALUE       | REFERENCES   |
|---------|-------------|--|
| ZF.DIA  | = 004000    | #5-50  |
| ZF.DLC  | = 000002    | #5-50  |
| ZF.DVP  | = 100000    | #5-50  |
| ZF.INI  | = 040000    | #5-50  |
| ZF.KMX  | = 000020    | #5-50  |
| ZF.LLC  | = 000004    | #5-50  |
| ZF.LMC  | = 000100    | #5-50  |
| ZF.MAN  | = 020000    | #5-50  |
| ZF.MFL  | = 000010    | #5-50  |
| ZF.MTM  | = 000400    | #5-50  |
| ZF.MUX  | = 000040    | #5-50  |
| ZF.PSE  | = 002000    | #5-50  |
| ZF.SLI  | = 010000    | #5-50  |
| ZF.TIM  | = 000200    | #5-50  |
| ZF.X3P  | = 000000    | #5-50  |
| ZS.ASN  | = 100000    | #5-50      20-540  |
| ZS.BSY  | = 140000    | #5-50      11-235  |
| Z.AVL   | 000014      | #5-50  |
| Z.DAT   | 000016      | #5-50  |
| Z.DSP   | 000000      | #5-50      5-50  |
| Z.FLG   | 000010      | #5-50  |
| Z.LEN   | = 000016    | #5-50  |
| Z.LLN   | 000006      | #5-50  |
| Z.MAP   | 000020      | #5-50  |
| Z.NAM   | 000004      | #5-50  |
| Z.PCB   | 000012      | #5-50  |
| Z.SCH   | 000007      | #5-50  |
| \$DLXPD | =        GX | 18-467   |
| \$PBLK  | =        GX | 10-165      10-173      11-198      11-227      11-236      11-237      11-241      13-298      13-309 |
|         |             | 13-316      14-334      15-359      15-367      15-368      16-396      17-438                         |



DLXLIN      CREATED BY    MACRO    ON 28-JUN-85 AT 22:55      PAGE 4      J 10

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |       |       |       |       |       |
|------------|------------|-------|-------|-------|-------|-------|
| CALL       | 6-69       | 7-146 | 7-149 | 7-166 | 7-173 | 8-201 |
| CALLR      | 7-180      |       |       |       |       | 8-211 |
| CCBDF\$    | #5-47      | 5-49  |       |       |       |       |
| MAP        | #5-46      |       |       |       |       |       |
| PDVDF\$    | #5-47      | 5-50  |       |       |       |       |
| PLADF\$    | #5-47      | 5-51  |       |       |       |       |
| RESRG      | #5-46      |       |       |       |       |       |
| RETURN     | 6-77       | 6-80  | 7-178 | 8-215 |       |       |
| SAVRG      | #5-46      |       |       |       |       |       |
| SLTDF\$    | #5-47      | 5-52  |       |       |       |       |
| \$ODLC     | #5-46      | 6-76  |       |       |       |       |

DLXQ10 - ISSUE 1/0 TO DLX  
Symbol table

MACRO V05.03b Saturday 29-Jun-85 12:23 Page 9-1

|                  |                 |                 |                 |                   |
|------------------|-----------------|-----------------|-----------------|-------------------|
| A\$CHK= 000000   | L\$VR2= 000010  | L\$NTI= 000200  | L\$IND 000126   | L\$TRB 000040     |
| A\$CPS= 000000   | L\$ZER= 000002  | L\$NTL= 000001  | L\$INI 000127   | L\$TYP 000063     |
| A\$PRI= 000000   | L\$MPT= 000010  | L\$NUM= 000100  | L\$LCT 000224   | L\$UNT 000003     |
| A\$TRP= 000000   | L\$MUX= 000004  | L\$OWN= 000010  | L\$LEN 000264   | L\$XCH 000232     |
| C\$CKP= 000000   | L\$NXC= 100000  | L\$PLT= 004000  | L\$LMB 000234   | L\$XMT 000120     |
| C\$ORE= 000400   | L\$TRB= 000002  | L\$PRO= 010000  | L\$LTM 000230   | L\$ASG= 000000    |
| C\$RSH= 177564   | L\$UNT= 000001  | L\$PVC= 020053  | L\$MRT 000253   | L\$DRV= 000000    |
| DEALUN= ***** GX | L\$WCN= 000040  | L\$RET= 000200  | L\$MSG 000056   | L\$PT1= 000001    |
| DLXERR 000020R   | L\$WDV= 000020  | L\$SER= 100000  | L\$MTYP 000075  | L\$11R= 000000    |
| DLXLUN 000200RG  | L\$WLD= 000360  | L\$STA= 000002  | L\$MWN 000252   | M\$CRB= 000124    |
| DLXMSG 000004R   | L\$WTR= 000200  | L\$SVC= 000362  | L\$MXB 000133   | M\$CRX= 000000    |
| DLXQIO 000026RG  | L\$WUN= 000100  | L\$STAD= 000020 | L\$MXR 000253   | M\$FCS= 000000    |
| DLXTXT 000005R   | L\$ACT= 000040  | L\$TH1= 000100  | L\$MXW 000252   | M\$MGE= 000000    |
| D\$BUG= 177514   | L\$BBT= 010000  | L\$TH2= 000200  | L\$NAM 000000   | M\$NET= 000000    |
| D\$ISK= 000000   | L\$BLK= 010000  | L\$TH3= 000400  | L\$NLEN= 000020 | M\$OVR= 000000    |
| D\$L11= 000001   | L\$BSA= 000001  | L\$XMT= 040000  | L\$NMT 000116   | N\$ACC= 000001    |
| D\$YNC= 000000   | L\$BSD= 000002  | L\$ACT 000104   | L\$NTI 000256   | N\$BUF= 000001    |
| D\$YNM= 000000   | L\$BSI= 000004  | L\$BBT 000114   | L\$NTL 000134   | N\$LDV= 000001    |
| E\$XPR= 000000   | L\$SCHN= 000001 | L\$BLK 000262   | L\$NUM 000240   | N\$MCP= 000001    |
| FMTLIN= ***** GX | L\$SCMB= 000002 | L\$BSA 000122   | L\$NUML 000236  | N\$MLL= 000001    |
| F\$LVL= 000001   | L\$SCOS= 000004 | L\$BSD 000123   | L\$NXT 000044   | N\$MOV= 000010    |
| G\$TTP= 000000   | L\$CUS= 000004  | L\$BSJ 000124   | L\$OPT 000062   | N\$NCT= 000001    |
| G\$TSS= 000000   | L\$DDT= 001000  | L\$BUF 000060   | L\$OWN 000102   | N\$PEM= 000001    |
| G\$TTK= 000000   | L\$DEA= 001000  | L\$CHN 000054   | L\$PAR 000072   | P\$P45= 000000    |
| G\$WRD= 000000   | L\$DLM= 001004  | L\$CMB 000234   | L\$PDV 000055   | P\$WRD= 000000    |
| G.LUCW= 000004   | L\$DLT= 002000  | L\$COS 000101   | L\$PFG 000041   | Q\$OPT= 000010    |
| G.LUFB= 000003   | L\$DTE= 000010  | L\$CTB 000053   | L\$PLT 000112   | R\$DER= 000000    |
| G.LUNA= 000000   | L\$HBT= 000400  | L\$CTL 000002   | L\$PRO 000075   | R\$K11= 000001    |
| G.LUNU= 000002   | L\$HTM= 010000  | L\$CUS 000260   | L\$PVC 000004   | R\$SND= 000000    |
| IE.LPN= ***** GX | L\$INA= 000010  | L\$DDM 000000   | L\$RET 000256   | R\$11M= 000000    |
| IOSB 000000R     | L\$IND= 000020  | L\$DDT 000106   | L\$SCN 000020   | S\$WRG= 000000    |
| I\$RAR= 000000   | L\$INJ= 000040  | L\$DEA 000105   | L\$SCR 000166   | S\$YSZ= 007600    |
| I\$RDN= 000000   | L\$LCT= 020000  | L\$DLT 000110   | L\$SER 000230   | T\$KMG= 000000    |
| K\$CNT= 177546   | L\$LMB= 000002  | L\$DTE 000240   | L\$LEN 000074   | T\$MIN= 000000    |
| K\$CSR= 177546   | L\$LOQ= 040000  | L\$DTEL 000236  | L\$SLT 000042   | V\$CTR= 001000    |
| K\$LDC= 000000   | L\$MDT= 020000  | L\$DTEP 000250  | L\$SNM 000046   | X\$DBT= 000000    |
| K\$TPS= 000074   | L\$MRT= 000020  | L\$FLG 000064   | L\$STA 000100   | \$CDSG= ***** GX  |
| L\$NTL= 000200   | L\$MWN= 000040  | L\$FLX 000076   | L\$TAD 000103   | \$DSW = ***** GX  |
| L\$OWN= 000400   | L\$MXB= 001000  | L\$FL1 000066   | L\$TH1 000130   | \$SAVAL= ***** GX |
| LDLP = 000000    | L\$MXR= 000020  | L\$FL2 000070   | L\$TH2 000131   | \$TMEFN= ***** GX |
| L\$MLT= 040000   | L\$MXW= 000040  | L\$HBT 000254   | L\$TH3 000132   | \$TMLUN= ***** GX |
| L\$REA= 000001   | L\$NMT= 020000  | L\$HTM 000226   | L\$TNM 000052   | \$SARG= 000002    |
| L\$SEG= 100000   | L\$NOR= 100000  | L\$INA 000125   | L\$TPT 000050   | \$SOST= 000014    |
| L\$SKP= 000004   |                 |                 |                 |                   |

. ABS. 000264 000 (RW,I,GBL,ABS,OVR)  
000266 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 13960 Words ( 55 Pages)  
Size of core pool: 15496 Words ( 59 Pages)  
Operating system: RSX-11M/PLUS

```

115      .SBTTL Validate optional characteristics buffer and relocate address
116
117      ;+
118      ;**BUFCHK-Validate optional characteristics buffer and relocate address
119      ;
120      ; Check the optional characteristics buffer supplied on transmit and
121      ; receive functions and relocate the address.
122
123      ; Inputs:
124      ; R3 = Address of I/O packet
125      ;       1.PRM+6 - Virtual address of optional characteristics buffer
126      ;       1.PRM+10- Length of optional characteristics buffer
127
128      ; Outputs:
129      ; R0 = Error code if buffer is invalid
130      ; R3 = Address of I/O packet
131      ;       1.PRM+6 - Bias of optional characteristics buffer
132      ;       1.PRM+10- Virtual address of optional characteristics buffer
133      ;       1.PRM+12- Length of optional characteristics buffer
134      ;       1.PRM+14- Virtual address of byte past end of characteristics buffer
135      ; 'C' Clear - Buffer is valid or not supplied
136      ; 'C' Set  - Buffer is invalid
137
138      ; Registers modified:
139      ; R0, R1, R2
140
141      BUFCHK: MOV     1.PRM+6(R3),R0 ; Get address of optional buffer
142              BNE     10$           ; If NE, present
143
144              CLR     1.PRM+10(R3) ; Clear remainder of
145              CLR     1.PRM+12(R3) ; buffer descriptor
146              CLR     1.PRM+14(R3)
147              RETURN
148
149      10$: MOV     1.PRM+10(R3),R1 ; Get buffer length
150              BEQ     20$           ; If EQ, illegal
151              CMP     R1,#8192.-64. ; Is it too large?
152              BHS     20$           ; If HS, yes
153
154              MOV     R1,1.PRM+12(R3) ; Return byte count to I/O packet
155              MOV     R1,1.PRM+14(R3) ; and for end of buffer address
156              CALL    @ACHK         ; Address check the buffer
157              BCS     20$           ; If CS, illegal buffer
158              CALL    @RELOC        ; Relocate buffer address
159              MOV     R1,1.PRM+6(R3) ; Store bias of optional buffer
160              MOV     R2,1.PRM+10(R3) ; and it's virtual address
161              ADD     R2,1.PRM+14(R3) ; Compute address of end of the buffer
162              MOV     #1.PRM+6,R0   ; Set offset to characteristics buffer
163              CALL    CHKCHR        ; Check buffer format
164              BCC     30$           ; If CC, buffer format is valid
165
166      20$: MOV     #1E.SPC&377,R0 ; Set up illegal buffer error code
167              SEC
168              RETURN
169      30$: RETURN

```



DLXSUB - DLX subroutines      MACRO V05.03b Friday 28-Jun-85 22:56 <sup>J 13</sup> Page 17-1  
Find free channel in process mapping table

698 001500                      90\$:    RESRG    <R5>                      ; Restore register  
699 001502                      RETURN

DLXSUB - DLX subroutines  
Symbol table

MACRO V05.03b Friday 28-Jun-85 22:56 Page 27-1

|                  |                 |                 |                 |                  |
|------------------|-----------------|-----------------|-----------------|------------------|
| ACHCK = ***** GX | CE.RTE= 100376  | CS.DEF= 000004  | C.XTC 000037    | FS.SFR= 006000   |
| ASSIGN 000000RG  | CE.RTL= 100003  | CS.DEV= 000002  | C.X25 000036    | FS.SFS= 004000   |
| ASSCHK= 000000   | CE.RTS= 100004  | CS.DIS= 000040  | DEASSN 000576RG | FS.SPW= 040000   |
| ASSCPS= 000000   | CE.SRC= 100364  | CS.ENA= 000001  | DSLIN 000716RG  | FS.STM= 000000   |
| ASSPRI= 000000   | CE.STP= 100352  | CS.ENB= 000020  | DSSBUG= 177514  | FS.STP= 002000   |
| ASSTRP= 000000   | CE.TME= 100354  | CS.ERR= 100000  | DSSISK= 000000  | FS.STR= 001000   |
| BLXIO = ***** GX | CE.TMO= 100374  | CS.FTL= 001000  | DSSL11= 000001  | FS.TRM= 003000   |
| BUFCHK 000104RG  | CE.UDF= 100001  | CS.HCR= 000001  | DSSYNC= 000000  | FS.WLB= 001000   |
| CAT5 000224RG    | CE.UNS= 100344  | CS.HFE= 002000  | DSSYNM= 000000  | FS.XKL= 002000   |
| CB.CCB= 000002   | CF.CHN= 000001  | CS.IGN= 000002  | D.BIAS 000000   | FS.XOF= 010000   |
| CB.DDM= 000040   | CF.EOM= 000004  | CS.LST= 040000  | D.SIZE 000004   | FS.XON= 007000   |
| CB.DLC= 000020   | CF.HDR= 000020  | CS.MTL= 004000  | ESSXPR= 000000  | FS.ZER= 002000   |
| CB.RDB= 000004   | CF.LB = 100000  | CS.RNG= 000010  | FC.CCP= 000020  | FS\$LV= 000001   |
| CB.SDB= 000010   | CF.LIN= 000002  | CS.ROV= 000004  | FC.CTL= 000006  | G\$SLN 002320RG  |
| CB.SLI= 000100   | CF.SOM= 000010  | CS.RSN= 010000  | FC.KCP= 000016  | G\$STPP= 000000  |
| CB.XLB= 000001   | CF.SYN= 000040  | CS.SHU= 000001  | FC.KIL= 000004  | G\$STSS= 000000  |
| CC.ADR= 000100   | CF.TRN= 000100  | CS.SID= 000002  | FC.MAN= 000024  | G\$STTK= 000000  |
| CC.DAD= 000102   | CHKCHR 000434RG | CS.STR= 000004  | FC.MLD= 000026  | G\$SWRD= 000000  |
| CC.DPA= 000302   | CLACT 000610RG  | CS.SUC= 000001  | FC.PCT= 000030  | IE.MOD= ***** GX |
| CC.DST= 000200   | CM.CIR= 000002  | CS.TMO= 020000  | FC.PWR= 000022  | IE.NOD= ***** GX |
| CC.ECM= 000311   | CM.DM1= 000253  | CS.XUR= 000004  | FC.RCE= 000002  | IE.NSF= ***** GX |
| CC.ECS= 000310   | CM.DM2= 000400  | CVTBL 000340RG  | FC.RCP= 000014  | IE.PRI= ***** GX |
| CC.ELS= 000304   | CM.DM3= 000000  | CS\$CKP= 000000 | FC.TIM= 000010  | IE.RSU= ***** GX |
| CC.EMC= 000300   | CM.FMT= 100000  | CS\$ORE= 000400 | FC.XCP= 000012  | IE.SPC= ***** GX |
| CC.EPH= 000303   | CM.HRD= 000002  | CS\$RSH= 177564 | FC.XME= 000000  | IS\$RAR= 000000  |
| CC.EPR= 000301   | CM.LIN= 000000  | C.ADD 000034    | FILCHN 000726RG | IS\$RDN= 000000  |
| CC.LCR= 000004   | CM.LOO= 000001  | C.BID 000003    | FINDC 001246RG  | I.PRM = ***** GX |
| CC.LDM= 000306   | CM.LO1= 000317  | C.BUF 000014    | FINDL 001262RG  | I.TCB = ***** GX |
| CC.LLC= 000200   | CM.LO2= 000000  | C.BUF1 000014   | FNDBR 001142RG  | KISAR6= ***** GX |
| CC.LSA= 000307   | CM.LO3= 000000  | C.BUF2 000024   | FNDFR 001350RG  | K\$SCNT= 177546  |
| CC.MCT= 000201   | CM.LT1= 000011  | C.CHRL 000010   | FNDLA 001504RG  | K\$SCSR= 177546  |
| CC.NET= 000000   | CM.LT2= 000053  | C.CNT 000020    | FNDLN 002026RG  | K\$SLDC= 000000  |
| CC.PDC= 000006   | CM.LT3= 007400  | C.CNT1 000020   | FNDPR 002254RG  | K\$STPS= 000074  |
| CC.PRO= 000101   | CM.RC1= 000253  | C.CNT2 000030   | FS.AST= 000000  | LDBGT = ***** GX |
| CC.SAD= 000010   | CM.RC2= 001000  | C.DAT1 000002   | FS.CIB= 002000  | LD\$LP = 000000  |
| CC.SID= 000305   | CM.RC3= 000000  | C.DATO 000004   | FS.CRA= 001000  | LF\$BRO= 002000  |
| CC.SPL= 000002   | CM.XLO= 000004  | C.FLG 000022    | FS.DIS= 013000  | LF\$CHN= 004000  |
| CC.STM= 000312   | COASN 000526RG  | C.FLG1 000022   | FS.DVC= 001000  | LF\$ENB= 000400  |
| CDTB 000346RG    | CP.CON= 001140  | C.FLG2 000032   | FS.ENB= 012000  | LF\$MOP= 001000  |
| CEMUL = ***** GX | CP.DCF= 000040  | C.FNC 000010    | FS.EXI= 001000  | LF\$RSR= 100000  |
| CE.ABO= 100362   | CP.DUM= 000540  | C.LIN 000006    | FS.GET= 006000  | LF.ACT= 100000   |
| CE.AC= 100012    | CP.HDL= 000007  | C.LNK 000000    | FS.HLT= 000000  | LF.BRO= 000400   |
| CE.DAO= 100346   | CP.LAT= 002140  | C.MOD 000011    | FS.INI= 000000  | LF.BWT= 000007   |
| CE.DIS= 100366   | CP.LOO= 000220  | C.NSP 000004    | FS.KIL= 000000  | LF.ENA= 002000   |
| CE.DNF= 100005   | CP.PS = 177400  | C.PRO 000042    | FS.LCL= 012000  | LF.LPB= 001000   |
| CE.ERR= 100370   | CP.PSI= 000200  | C.RSV 000002    | FS.LTM= 001000  | LF.MDC= 000100   |
| CE.ILC= 100006   | CP.ROU= 001540  | C.STA 000007    | FS.MNT= 004000  | LF.MFL= 004000   |
| CE.ILN= 100350   | CP.XCF= 000100  | C.STAT 000006   | FS.MSN= 014000  | LF.MTP= 000020   |
| CE.INV= 100002   | CP.2FR= 000030  | C.STS 000012    | FS.REA= 001000  | LF.PAC= 000200   |
| CE.IUM= 100013   | CS.ABO= 000100  | C.TYP 000000    | FS.RET= 000000  | LF.RDY= 040000   |
| CE.LTO= 100356   | CS.BRO= 000002  | C.URM 177776    | FS.REZ= 003000  | LF.REA= 010000   |
| CE.MCE= 100007   | CS.BUF= 000200  | C.XACP 000004   | FS.RLB= 002000  | LF.SER= 000040   |
| CE.MOP= 100372   | CS.CES= 000002  | C.XID 000035    | FS.RNG= 011000  | LF.TIM= 000010   |
| CE.NMA= 100014   | CS.CHN= 000010  | C.XLEN 000044   | FS.RST= 000000  | LF.UNL= 020000   |
| CE.NTE= 100361   | CS.CMP= 000200  | C.XPLI 000040   | FS.RTN= 001000  | LF.X2P= 000000   |
| CE.PCN= 100011   | CS.DAO= 000003  | C.XPT 000034    | FS.SET= 005000  | LLCTA = ***** GX |
| CE.RES= 100010   | CS.DCR= 000400  | C.XSVC 000042   | FS.SFC= 005000  | LN.CLO= 000300   |

EVLE - EVENT LOGGER PROCESS      MACRO V05.03b Friday 28-Jun-85 22:58 <sup>J 15</sup> Page 7  
DISPATCH TABLE FOR COMM/EXEC ENTRY

|    |        |         |           |                                    |        |
|----|--------|---------|-----------|------------------------------------|--------|
| 75 |        |         | .SBTTL    | DISPATCH TABLE FOR COMM/EXEC ENTRY |        |
| 76 |        |         |           |                                    |        |
| 77 | 000000 | 000042' | \$EVLTB:: | .WORD                              | NELOG  |
| 78 | 000002 | 000003' |           | .WORD                              | .+1    |
| 79 | 000004 | 000005' |           | .WORD                              | .+1    |
| 80 | 000006 | 000007' |           | .WORD                              | .+1    |
| 81 | 000010 | 000426' |           | .WORD                              | NETIME |

; LOG NETWORK EVENT  
; TRANSMIT ENABLE (ILLEGAL)  
; KILL ENABLE (ILLEGAL)  
; CONTROL ENABLE (ILLEGAL)  
; TIMEOUT

```

Symbol table
FS.RNG= 011000 KISAR6= ***** GX L.OWNR 000021 QC$CIP= 000001 RE$OPR= 000000
FS.RST= 000000 KSAR6 000020R L.UNT 000013 QC$ENB= 000001 RE$RCV= 000001
FS.RTN= 001000 K$CNT= 177546 MAPCHN 000636R QC$STP= 000002 RE$SED= 000011
FS.SET= 005000 K$CSR= 177546 MAPORT 000702R QC$SVC= 000002 RE$SKW= 000006
FS.SFC= 005000 K$SLDC= 000000 M$SAC= 000016 QF$EXT= 000377 RE$STA= 000002
FS.SFR= 006000 K$STPS= 000074 M$SPR= 000012 QF$FAS= 000001 RE$SUM= 000003
FS.SFS= 004000 LD$LP= 000000 M$SSV= 000014 QF$NRS= 000144 RE$SSN= 000000
FS.SPW= 040000 LF.ACT= 100000 M$25A= 000006 QF$PKS= 000102 RE$TME= 000021
FS.STM= 000000 LF.BRO= 000400 M$25P= 000002 QF$RRS= 000300 RE$TMO= 000000
FS.STP= 002000 LF.BWT= 000007 M$25S= 000004 QF$TPC= 000002 RE$TMR= 000020
FS.STR= 001000 LF.ENA= 002000 M$29S= 000010 QF$WSZ= 000103 RE$UPT= 000002
FS.TRM= 003000 LF.LPB= 001000 M$HIGH= 000003 Q$SACT= 000016 RE$URE= 000003
FS.WLB= 001000 LF.MDC= 000100 M$LCRB= 000124 Q$APV= 000002 RE$VER= 000005
FS.XKL= 002000 LF.MFL= 004000 M$CRX= 000000 Q$CCO= 000030 RE$VRQ= 000015
FS.XOF= 010000 LF.MTP= 000020 M$FCFS= 000000 Q$CLC= 000022 RT$INI= 000002
FS.XON= 007000 LF.PAC= 000200 M$MGE= 000000 Q$CLR= 000026 RT$OFF= 000001
FS.ZER= 002000 LF.RDY= 040000 M$NET= 000000 Q$CRQ= 000004 RT$ON = 000000
F$LLVL= 000001 LF.REA= 010000 M$QVR= 000000 Q$CRV= 000020 R$SDER= 000000
G$STPP= 000000 LF.SER= 000040 M$3100= 000000 Q$CWC= 000024 R$SK11= 000001
G$STSS= 000000 LF.TIM= 000010 M$3101= 000001 Q$DAT= 000000 R$SND= 000000
G$STTK= 000000 LF.UNL= 020000 M$3102= 000002 Q$DPV= 000032 R$11M= 000000
G$SWRD= 000000 LF.X2P= 000000 M$3103= 000003 Q$RCO= 000012 SC$OFF= 000001
HF$DLM= 000002 LGDDB 000022R NELOG 000042R Q$RSC= 000004 SC$ON = 000000
HF$GWY= 000010 LGPDV 000024R NETIME 000426R Q$RSR= 000002 SC$RST= 000003
HF$HOS= 000004 LGSTT 000026R N$SACC= 000001 Q$RST= 000010 SC$SHU= 000002
HF$XDF= 000020 LN.CLO= 000000 N$BUB= 000001 Q$RWC= 000006 SF.ACT= 000200
H$CUG 000010 LN.DUM= 000005 N$LDV= 000001 Q$WAC= 000014 SF.ENA= 000100
H$DST 000012 LN.LOA= 000004 N$MCP= 000001 Q$WCA= 000000 SF.LPB= 000004
H$D29 000014 LN.LOO= 000003 N$MLL= 000001 QUEUE 001132R SF.MFL= 000040
H$FLG 000000 LN.OAU= 000003 N$MOV= 000010 Q$ALNK 000004 SF.PAC= 000020
H$GLEN 000104 LN.OFF= 000001 N$NCT= 000001 Q$BUF2 000020 SF.REA= 000010
H$GLT 000044 LN.ON = 000000 N$PEM= 000001 Q$CB 000012 SF.SER= 000001
H$GNAM 000050 LN.OOP= 000004 OC$ACK= 000001 Q$CBO 000014 SF.SVC= 000002
H$GNML= 000020 LN.OPE= 000001 OP$INI= 000000 Q$CNTL 000007 SF.UNL= 000040
H$GPT 000046 LN.RFE= 000002 Q$FLG 000006 SLTMA 000036R
H$HITS 000034 LN.SER= 000002 Q$ACHD 000012 SV$DUM= 000001
H$HLEN 000044 LN.STA= 000017 Q$CALQ 000016 Q$LNK 000000 SV$LOA= 000000
H$LBDA 000070 LN.SUB= 000360 Q$CLK 000000 Q$PORT 000003 S$SWG= 000000
H$LBND 000072 LN.TRT= 000006 Q$CNTL 000003 Q$SBST 000010 S$YSZ= 007600
H$LDTE 000002 L$SAG= 000000 Q$LENG 000022 Q$STE 000011 S.COST 000001
H$LEN 000042 L$DRV= 000000 Q$LINE 000004 Q$STN 000002 S.FLG 000000
H$LOTS 000032 L$SP11= 000001 Q$TRHD 000006 Q$TIMZ 000036 S.LEN 000004
H$NETW 000024 L$11R= 000000 Q$WCT 000002 Q$XABQ 000024 S.NMST 000002
H$NML = 000006 L.COST 000015 P$HDE= 000004 Q$XPLI 000034 S.OWNR 000003
H$NPT 000022 L.CTL 000012 P$LOC= 000002 Q$XWTQ 000030 TTNS 000012R
H$PTB 000020 L.CVA 177776 P$MTS= 000003 Q$OPT= 000010 T$SKMG= 000000
H$PVC 000006 L.DDM 000002 P$UMP= 000000 RE$ADC= 000004 T$MIN= 000000
H$RDTE 000004 L.DDS 000004 P$WCS= 000001 RE$ADF= 000017 V$CTR= 001000
H$RNW 000042 L.DLC 000003 PR7 = ***** GX RE$ADR= 000007 X$NCO= 000020
H$SVC 000036 L.DLM 000006 PS = ***** GX RE$BLK= 000010 X$RR = 000200
H$TRB 000016 L.DLS 000010 P$IPT 000034R RE$CAF= 000014 X$UCL= 000040
H$XAVL 000100 L.FLG 000000 P$P45= 000000 RE$DAT= 000001 X$URE= 000100
H$XBIA 000074 L.KRBA 000016 P$WRD= 000000 RE$DRP= 000016 X$NCO= 000020
H$X29C 000040 L.LEN = 000022 QC$ACK= 000004 RE$LDT= 000013 X$OFF= 000200
INIFLG 000530R L.MPF 000022 QC$ACT= 000004 RE$LSN= 000012 X$ORI= 000002
I$RAR= 000000 L.NMST 000020 RE$NML= 000001 X$OTI= 000001
I$RDN= 000000 L.NSTA 000014 RE$OPE= 000004 X$TIL= 000010

```

```

330 .SBTTL Build image mode field
331
332 :+
333 :*-BLDIMG-Build image mode field
334 :
335 :   Build an image mode field in the message buffer after checking for
336 :   valid size.
337 :
338 :   Calling sequence:
339 :       JSR      R3,BLDIMG
340 :       .WORD    <Maximum field size>
341 :
342 :   Inputs:
343 :       R0 = Pointer to count field in connect block
344 :       R2 = Pointer to next available byte in message buffer
345 :
346 :   Outputs:
347 :       R2 = Pointer past last byte used in message buffer
348 :       'C' Clear - Valid image field built
349 :       'C' Set - Format error
350 :
351 :   Registers modified:
352 :       R0, R1
353 :
354 :BLDIMG: MOV      (R0)+,R1      ; Get length of the field
355 :          CMP      (R3)+,R1    ; Is the field too large?
356 :          BLO      10$         ; If LO, yes (C-bit set)
357 :          CALL     MOVIMG      ; Move image field
358 :
359 :10$:   RTS      R3

```

SESUSR - Session control user i MACRO V05.03b Friday 28-Jun-85 20:01<sup>K</sup> Page 20-3

Symbol table

|          |           |    |          |          |         |        |          |          |           |          |
|----------|-----------|----|----------|----------|---------|--------|----------|----------|-----------|----------|
| SAVOP=   | *****     | GX | SSPSEG=  | 000100   | S.EUNN= | 100376 | T\$NVR   | 000001   | W.CINT    | 000022   |
| SETREQ   | 000000R   |    | SSREFJ   | = 000004 | S.EURO= | 100374 | T\$RPRI  | 000040   | W.CSND    | 000020   |
| SNDACC   | 0000056RG |    | SSSND    | = 000006 | S.SEOM= | 000003 | T\$SYC   | 000034   | W.CTL     | 000000   |
| SNDCON   | 000170RG  |    | SSSNI    | = 000014 | S.SSUC= | 000001 | T\$T5    | 000030   | W.KAST    | 000014   |
| SNDLIS   | 0001300RG |    | SSMDA    | = 000002 | TRMLNK= | *****  | T\$T6    | 000032   | W.LLT     | 000004   |
| SNSESD=  | *****     | GX | SS\$WRG= | 000000   | T\$FLAG | 000044 | T\$SKMG= | 000000   | W.LUN     | 000003   |
| SiCC     | = 000004  |    | SS\$YSZ= | 007600   | T\$LI   | 000013 | T\$MIN=  | 000000   | W.MBOX    | 000012   |
| ST\$CIR= | 000006    |    | S.EABL   | = 100200 | T\$LI   | 000013 | T.UCB    | = *****  | W.RCVQ    | 000024   |
| ST\$CIS= | 000002    |    | S.EABM=  | 100370   | T\$LI   | 000013 | UISARG=  | *****    | W.SEGZ    | 000006   |
| ST\$DAT= | 000010    |    | S.EABO=  | 100367   | T\$LI   | 000013 | USINT    | 001346RG | W.SNDQ    | 000016   |
| ST\$DIP= | 000012    |    | S.EABS=  | 100202   | T\$LIN  | 000000 | US\$CNF= | 000002   | W.STAT    | 000002   |
| ST\$PND= | 000014    |    | S.EACR=  | 100336   | T\$LI   | 000006 | US\$DIS= | 000006   | W.TMP     | 000010   |
| SSABO    | = 000022  |    | S.ECBE=  | 100204   | T\$LLD  | 000012 | US\$DON= | 000000   | W.WBL     | 000026   |
| SSACC    | = 000002  |    | S.EDBO=  | 100206   | T\$LLDC | 000045 | US\$DSC= | 000004   | X\$SDBT=  | 000000   |
| SSBUF    | = 000026  |    | S.EERR=  | 100210   | T\$LLDL | 000012 | US\$EAC= | 000012   | \$ACCNT=  | *****    |
| SS\$CNR  | = 000000  |    | S.EIDM=  | 100214   | T\$LLDO | 000012 | US\$WDS= | 000010   | \$CALLX=  | *****    |
| SSCON    | = 000000  |    | S.EINF=  | 100212   | T\$LLDS | 000012 | VE.FAI   | = 177777 | \$SDDFM=  | *****    |
| SSDAT    | = 000002  |    | S.EIOF=  | 100373   | T\$LLEN | 000046 | VS.NPV   | 000001   | \$DSMOD=  | *****    |
| SSDLS    | = 000020  |    | S.ELNS   | = 100365 | T\$LOPR | 000002 | VS.PRIV  | 000002   | \$FLAGS=  | *****    |
| SSDRQ    | = 000012  |    | S.ELST=  | 100216   | T\$LTCL | 000024 | VZ.NVD   | = 000000 | \$FLOW    | = *****  |
| SSDSR    | = 000006  |    | S.ELWS   | = 100220 | T\$LTIM | 000026 | V\$CTR   | = 001000 | \$IOPKT   | = *****  |
| SSGLN    | = 000024  |    | S.EMTL   | = 100222 | T\$LTPR | 000014 | WK.ACK   | = 000001 | \$LTM     | = *****  |
| SSINT    | = 000004  |    | S.ENOF   | = 100224 | T\$LTPS | 000020 | WK.AST   | = 000200 | \$MAIBX   | = *****  |
| SSIRQ    | = 000016  |    | S.ENRO   | = 100332 | T\$NAPL | 000004 | WK.DIS   | = 000010 | \$PASSW   | = *****  |
| SSMRQ    | = 000010  |    | S.ENSL   | = 100327 | T\$NFE  | 000000 | WK.INT   | = 000020 | \$REASN   | = *****  |
| SSNCT    | = 000010  |    | S.ENUR   | = 100331 | T\$NLEN | 000010 | WK.RCV   | = 000004 | \$REQID   | = *****  |
| SSNTIF   | = 000001  |    | S.EOTB   | = 100372 | T\$NNUL | 000002 | WK.SND   | = 000002 | \$SRDFM   | = *****  |
| SSPBOM   | = 000040  |    | S.ERBO   | = 100226 | T\$NOPL | 000006 | WS.DIP   | = 000010 | \$SRVCS   | = *****  |
| SSPEOM   | = 000100  |    | S.ERES   | = 100377 | T\$NRNI | 000042 | WS.INT   | = 000002 | \$\$SHFT  | = 000001 |
| SSPMOM   | = 000000  |    | S.ERNS   | = 100375 | T\$NRPL | 000005 | WS.KAS   | = 000004 | .\$\$\$\$ | = 000034 |
| SSPMSG   | = 000200  |    | S.ETMI   | = 100230 | T\$NRUL | 000007 | WS.STA   | = 000001 |           |          |

```

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)
      001464 001 (RW,I,LCL,REL,CON)
$HIGH 000136 002 (RW,I,LCL,REL,CON)
Errors detected: 0

```

```
*** Assembler statis:  S
```

```
Work file reads: 126
Work file writes: 127
Size of work file: 29925 Words ( 117 Pages)
Size of core pool: 17608 Words ( 67 Pages)
Op-erating system: RSX-11M/PLUS
```

Elapsed time: 00:00:39.25  
SY:SESUSR11S.V2.[131,134]SESUSR11S/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCS/PA:1,[131,10]V2,SESUSR

```

77                                     .SBTTL Control enable processing
78                                     ;+
79                                     ;**--CTLEN-Control enable processing
80                                     ;
81                                     ;   This routine is entered from the COMM/EXEC when a control enable
82                                     ;   request is dispatch to this process.
83                                     ;-
84                                     ; Inputs:
85                                     ;   R3 = Subfunction code
86                                     ;   R4 = Address of the control enable CCB
87                                     ;
88 000022 CTLEN:: CALL   STLNK      ; Compute address of the physical link access block
89 000026      CALLR @CTLDSP(R3) ; Dispatch to processing routine
90
91 000032 000033' CTLDSP: .WORD  .+1      ; Reserved
92 000034 000040'      .WORD  START    ; Start or line assignment
93 000036 000132'      .WORD  SHUT     ; Shutdown

```

|     |        |        |        |               |                    |                                       |
|-----|--------|--------|--------|---------------|--------------------|---------------------------------------|
| 538 | 001320 | 005065 | 000012 | CLR           | L\$CTL(R5)         | ; None present now                    |
| 539 | 001324 | 116500 | 000001 | MOVB          | L\$FLG(R5),R0      | ; Get current state of Link enable    |
| 540 | 001330 | 042700 | 177776 | BIC           | #*C<L\$ENB/400>,R0 |                                       |
| 541 | 001334 | 042715 | 000400 | BIC           | #L\$ENB,(R5)       | ; Lose all knowledge of enable state  |
| 542 | 001340 |        |        | CALLR         | CCBDN              | ; Complete the stop request           |
| 543 |        |        |        |               |                    |                                       |
| 544 | 001344 | 016546 | 000014 | 80\$: MOV     | L\$FNC(R5),-(SP)   | ; Save pending control function queue |
| 545 | 001350 | 005065 | 000014 | CLR           | L\$FNC(R5)         | ; and reset the listhead              |
| 546 |        |        |        |               |                    |                                       |
| 547 | 001354 | 012603 |        | 90\$: MOV     | (SP)+,R3           | ; Get next I/O packet                 |
| 548 | 001356 | 001404 |        | BEQ           | 100\$              | ; If EQ, all done                     |
| 549 | 001360 | 011346 |        | MOV           | (R3),-(SP)         | ; Save address of next I/O packet     |
| 550 | 001362 |        |        | CALL          | NXINA              | ; and reinitiate it                   |
| 551 | 001366 | 000772 |        | BR            | 90\$               |                                       |
| 552 |        |        |        |               |                    |                                       |
| 553 | 001370 |        |        | 100\$: RETURN |                    |                                       |
| 554 |        |        |        |               |                    |                                       |
| 555 |        | 000001 |        |               | .END               |                                       |



```

105 .SBTTL Retrieve Ethernet address
106
107 :+
108 **--RCVADR-Retrieve Ethernet address
109 Retrieve the Ethernet source address for this received message.
110
111 :
112 Inputs:
113 R0 = Address of characteristics block
114 R1 = Default status (CS.IGN = Characteristic ignored)
115 R3 = Address of the I/O packet
116 R4 = Address of the receive CCB
117 C.BUF - Bias of received message
118 C.BUF+2 - Virtual address of received message
119 C.CNT - Length of received message
120 C.NSP - Virtual address of Ethernet header
121 C.PRO - Protocol type from message
122 C.ADD - Source address from Ethernet message
123 R5 = Address of the physical link access block
124
125 :
126 Outputs:
127 R1 = Status to return for characteristic
128
129 RCVADR: CMP C.DATI(R0),#6 ; Are there at least 6 bytes available?
130 BLO 10$ ; If LO, no
131
132 BIT #LFBRO,(R5) ; Is this a broadcast channel?
133 BEQ 10$ ; If EQ, no
134
135 MOV C.ADD(R4),C.CHRL(R0)
136 MOV C.ADD+2(R4),C.CHRL+2(R0)
137 MOV C.ADD+4(R4),C.CHRL+4(R0)
138 MOV #6,C.DATO(R0) ; Fill in Ethernet address and it's size
139 MOV #CS.SUC,R1 ; Return successful completion
140
141 10$: RETURN

```

44  
45  
46  
47  
48  
49 000000  
50 000000  
51 000000  
52 000000

.SBTTL Macro definitions

.MCALL SAVRG,RESRG,MAP  
.MCALL CCBDF\$,PDVDF\$,PLADF\$,SLTDF\$

CCBDF\$ ; Define CCB offsets  
PDVDF\$ ; Define PDV offsets  
PLADF\$ ; Define PLA offsets  
SLTDF\$ ; Define SLT offsets

```

401      .SBTTL Set network management state for <Line-id>
402
403      +
404      **--CTLLST-Set network management state for <Line-id>
405      Set up the network management state field of the network management
406      word for the <Line-id> to the specified process.
407
408      -
409      Inputs:
410      R1 = Address of system line table
411      R3 = Address of I/O packet
412      I.PRM+6 - Network management state
413
414      Registers modified:
415      R0, R1, R2, R3, R4, R5
416
417      .ENABL LSB
418 000646 012704 000017      CTLLST: MOV    #LN.STA,R4      ; Set up mask for bit field
419 000652 000402              BR      10$          ; Enter common code
420
421      .SBTTL Set network management substate for <Line-id>
422
423      +
424      **--CTLLST-Set network management substate for <Line-id>
425      Set up the network management substate field of the network management
426      word for the <Line-id> to the specified process.
427
428      -
429      Inputs:
430      R1 = Address of system line table
431      R3 = Address of I/O packet
432      I.PRM+6 - Network management substate
433
434      Registers modified:
435      R0, R1, R2, R3, R4, R5
436 000654 012704 000360      CTLSST: MOV    #LN.SUB,R4      ; Set up mask for bit field
437
438 000660 016702 000010G      10$:  MOV    $PBLK+T$SLN,R2    ; Get SLN and tributary pair
439 000664              CALL   STNMST          ; Compute address of network management byte
440 000670 040415              BIC    R4,(R5)      ; Clear out the bit field
441 000672 016302 000006G      MOV    I.PRM+6(R3),R2    ; Get new bit field value
442 000676 005104              COM     R4          ; Isolate new bit field
443 000700 040402              BIC    R4,R2      ;
444 000702 050215              BIS    R2,(R5)    ; Set new network management state/substate
445 000704              CALLR   ISSUC          ; Complete the request successfully
446
447      .DSABL LSB

```

DLXCTL      CREATED BY    MACRO    ON 28-JUN-85 AT 22:54      PAGE 5      K 8  
 MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES                |                           |                           |                            |                            |                  |                  |                  |                  |                  |
|------------|---------------------------|---------------------------|---------------------------|----------------------------|----------------------------|------------------|------------------|------------------|------------------|------------------|
| CALL       | 9-135<br>13-303<br>20-539 | 9-141<br>13-306<br>21-567 | 9-144<br>13-311<br>22-585 | 10-168<br>14-337<br>22-586 | 10-170<br>14-339<br>23-614 | 10-174<br>15-362 | 11-224<br>16-391 | 11-229<br>16-397 | 11-232<br>17-439 | 13-301<br>20-537 |
| CALLR      | 9-149<br>23-616           | 11-244                    | 12-279                    | 13-318                     | 15-371                     | 16-399           | 17-445           | 18-472           | 19-504           | 20-547           |
| CCBDF\$    | #5-47                     | 5-49                      |                           |                            |                            |                  |                  |                  |                  |                  |
| DISP       | #6-56                     | 19-509                    | 19-510                    | 19-511                     | 19-512                     | 19-513           |                  |                  |                  |                  |
| MAP        | #5-46                     |                           |                           |                            |                            |                  |                  |                  |                  |                  |
| PDVDF\$    | #5-47                     | 5-50                      |                           |                            |                            |                  |                  |                  |                  |                  |
| PLADF\$    | #5-47                     | 5-51                      |                           |                            |                            |                  |                  |                  |                  |                  |
| RESRG      | #5-46                     | 9-145                     |                           |                            |                            |                  |                  |                  |                  |                  |
| SAVRG      | #5-46                     | 9-140                     |                           |                            |                            |                  |                  |                  |                  |                  |
| SLTDF\$    | #5-47                     | 5-52                      |                           |                            |                            |                  |                  |                  |                  |                  |

DLXLIN - DLX Line control routi MACRO V05.03b Friday 28-Jun-85 22:54<sup>K 9</sup>  
Table of contents

|    |     |                                       |
|----|-----|---------------------------------------|
| 5- | 44  | Macro definitions                     |
| 6- | 54  | Issue control request to DLC          |
| 7- | 82  | Stop operation of a physical channel  |
| 8- | 184 | Start operation of a physical channel |

```

DDDDDDDD    LL      XX      XX      QQQQQQ      IIIII      000000
DDDDDDDD    LL      XX      XX      QQQQQQ      IIIII      000000
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DD      DD   LL      XX      XX      QQ      QQ      II      00      00
DDDDDDDD    LLLLLLLLLL  XX      XX      QQQQ      QQ      IIIII      000000
DDDDDDDD    LLLLLLLLLL  XX      XX      QQQQ      QQ      IIIII      000000

```

```

      SSSSSSSS   TTTT TTTT TTT
      SSSSSSSS   TTTT TTTT TTT
          SS           TT
          SS           TT
          SS           TT
          SS           TT
              SSSSSS   TT
              SSSSSS   TT
                      SS   TT
                      SS   TT
                      SS   TT
                      SS   TT
          SSSSSSSS   TT
          SSSSSSSS   TT
LLLLLLLLLLLLL
LLLLLLLLLLLLL

```

DLXQIO - ISSUE I/O TO DLX      MACRO V05.03b Saturday 29-Jun-85 12:23 Page 9-2  
Symbol table

Elapsed time: 00:00:16.05  
SY:DLXQIO.V2,[135,134]DLXQIO/CR/~SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[135,10]DLXQIO

```
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186 000224
187 000230 005001
188 000232 012702 000340'
189
190 000236 112005
191 000240 020400
192 000242 103431
193
194 000244 122705 000132
195 000250 103424
196 000252 122705 000101
197 000256 101021
198
199 000260 162705 000100
200 000264
201 000270 012200
202 000272 010501
203 000274
204 000300 060116
205 000302
206
207 000306 020227 000346'
208 000312 103751
209 000314 112005
210 000316 020400
211 000320 103402
212
213 000322 120305
214 000324 001001
215 000326 005727
216 000330 000261
217 000332
218 000336
219
220
221
222
223 000340 003100
224 000342 000050
225 000344 000001
```

```
.SBTTL Convert ASCII to RAD50
;+
;--CAT5-Convert ASCII to RAD50
; Convert an ASCII process name to RAD50 (3 characters).
; Inputs:
; R0 = Address of first byte in the <Line-id> string
; R3 = Legal terminating character
; R4 = Address of the end of the string +1
; Outputs:
; R0 = Address of next byte in the <Line-id> string
; R1 = Converted result
; 'C' Clear - Successful conversion
; 'C' Set - Illegal character in the string
CAT5:: SAVRG <R2,R5> ; Get some free registers
      CLR R1 ; Clear accumulation
      MOV #CVTBL,R2 ; Point to conversion table
10$: MOVB (R0)+,R5 ; Get next character
      CMP R4,R0 ; Reached end of the string yet?
      BLO 30$ ; If LO, yes
      CMPB #'Z,R5 ; Is it alphabetic, below Z?
      BLO 20$ ; If LO, no
      CMPB #'A,R5 ; Is it above A?
      BHI 20$ ; If HI, no
      SUB #100,R5 ; Scale to RAD50
      SAVRG <R0,R1> ; Save some registers
      MOV (R2)+,R0 ; Get next scale factor
      MOV R5,R1 ; Copy character
      CALL @CEMUL ; Multiply it out
      ADD R1,(SP) ; Accumulate result
      RESRG <R1,R0> ; Recover registers
      CMP R2,#CVTBL+6 ; Have we converted enough characters?
      BLO 10$ ; If LO, no
      MOVB (R0)+,R5 ; Get terminating character
      CMP R4,R0 ; Reached the end of the string?
      BLO 30$ ; If LO, yes
20$: CMPB R3,R5 ; Is this a legal terminator?
      BNE 40$ ; If NE, no
30$: TST (PC)+ ; Return success status
40$: SEC ; Indicate illegal character seen
      RESRG <R5,R2> ; Restore registers
      RETURN
;+
; Conversion control table
CVTBL: .WORD 50*50
      .WORD 50
      .WORD 1
```



```

701      .SBTTL Scan databases for <Line-id>
702
703      ;+
704      ;**--FNDLA-Scan databases for <Line-id>
705
706      Using the contents of the common parameter block, scan the COMM/EXEC
707      databases for the specified <Line-id> and return further information
708      in the parameter block.
709
710      -
711      Outputs:
712      R1 = DDM: Address of system line table
713      R1 = LLC: Address of process PDV
714      'c' Clear - <Line-id> found in databases
715      'c' Set - <Line-id> not present in databases
716
717      Registers modified:
718      R0, R2
719
720      FNDLA:: MOV $PBLK+T$NAM,R2 ; Get RAD50 process name
721      CALL $PDVID ; Find the process' PDV index
722      BCS 100$ ; If CS, no process of this name
723
724      MOV $PBLK+T$PDV ; Save process' PDV index
725      ADD $PDVTA,R2 ; Compute address within PDV index table
726      MOV (R2),R1 ; Get address of PDV
727      MOV $PBLK+T$FLG,R0 ; Get process flag byte
728      MOV $PBLK+T$FLG,R0 ; and save it
729
730      BITB #ZF.LLC,R0 ; Is process an LLC?
731      BNE 90$ ; If NE, yes
732      BITB #ZF.DDM,R0 ; Is process a DDM?
733      BEQ 90$ ; If EQ, no ... all done
734
735      MOV @SLTNM,R0 ; Get # of system lines
736      MOV @SLTMA,R2 ; Point to system line index table
737
738      10$: MOV (R2)+,R1 ; Get address of system line table entry
739      BEQ 20$ ; If EQ, none present
740      CMPB L.DDM(R1),$PBLK+T$PDV ; If NE, not this process
741      BNE 20$ ; If NE, not this controller
742      CMPB L.CTL(R1),$PBLK+T$CTL ; If NE, not this controller
743      BEQ 40$ ; If EQ, we have a match
744
745      20$: SOB R0,10$ ; Check all system lines
746      30$: SEC ; Indicate <Line-id> not present
747      BR 100$ ; and exit
748
749      40$: MOV (R1),$PBLK+T$SLF ; Fill in parameter block
750      MOV L.DDM(R1),$PBLK+T$DDM ; ...
751      MOV L.KRBA(R1),$PBLK+T$KRB ; ...
752      MOV L.NMST(R1),$PBLK+T$NMST ; ...
753
754      MOV @SLTNM,R2 ; Compute the system line number
755      SUB R0,R2 ; ...
756      MOV $PBLK+T$SLN,R2 ; ...
757
```

DLXSUB - DLX subroutines  
Symbol table

MACRO V05.03b Friday 28-Jun-85 22:56 Page 27-2

K 14

|                 |                  |                  |                   |                    |
|-----------------|------------------|------------------|-------------------|--------------------|
| LN.DUM= 000005  | L\$STT 000007    | N\$SACC= 000001  | STLNA 002464RG    | ZF.DIA= 004000     |
| LN.LOA= 000004  | L\$TCB 000022    | N\$SBUF= 000001  | STLNK 002460RG    | ZF.DLC= 000002     |
| LN.LOO= 000003  | L\$TIM 000004    | N\$SLDV= 000001  | STNMST 002516RG   | ZF.DVP= 100000     |
| LN.OAU= 000003  | L\$TIMJ 000005   | N\$SMCP= 000001  | STSLT 002602RG    | ZF.INI= 040000     |
| LN.OFF= 000001  | L\$TIP 000003    | N\$SMML= 000001  | S\$SWRG= 000000   | ZF.KMX= 000020     |
| LN.ON = 000000  | L\$XMT 000016    | N\$SMOV= 000010  | S\$YSZ= 007600    | ZF.LLC= 000004     |
| LN.OOP= 000004  | L\$ASG= 000000   | N\$SNCT= 000001  | S.COST 000001     | ZF.LMC= 000100     |
| LN.OPE= 000001  | L\$DRV= 000000   | N\$SPEM= 000001  | S.FLG 000000      | ZF.MAN= 020000     |
| LN.REF= 000002  | L\$SP11= 000001  | PDVID = ***** GX | S.LEN 000004      | ZF.MFL= 000010     |
| LN.SER= 000002  | L\$11R= 000000   | PDVTA = ***** GX | S.NMST 000002     | ZF.MTM= 000400     |
| LN.STA= 000017  | L.COST 000015    | PLAST = ***** GX | S.OWNR 000003     | ZF.MUX= 000040     |
| LN.SUB= 000360  | L.CTL 000012     | PRCLD 002416RG   | T\$ASN 000016     | ZF.PSE= 002000     |
| LN.TRI= 000006  | L.CVA 177776     | P\$P45= 000000   | T\$CTL 000002     | ZF.SLI= 010000     |
| LR\$CTL= 000003 | L.DDM 000002     | P\$WRD= 000000   | T\$DDM 000014     | ZF.TIM= 000200     |
| LR\$DEA= 000200 | L.DDS 000004     | Q\$OPT= 000010   | T\$DLC 000015     | ZF.X3P= 000000     |
| LR\$DIS= 000020 | L.DLC 000003     | RBRT = ***** GX  | T\$FLG 000006     | ZS.ASN= 100000     |
| LR\$STP= 000010 | L.DLM 000006     | RELOC = ***** GX | T\$KRB 000020     | ZS.BSY= 140000     |
| L\$SDAT= 000004 | L.DLS 000010     | RLSCHN 002444RG  | T\$LEN 000024     | Z.AVL 000014       |
| L\$ERR= 000200  | L.FLG 000000     | R\$DER= 000000   | T\$LIN 000003     | Z.DAT 000016       |
| L\$FRE= 000000  | L.KRBA 000016    | R\$K11= 000001   | T\$NAM 000000     | Z.DSP 000000       |
| L\$IDL= 000001  | L.LEN = 000022   | R\$SND= 000000   | T\$NMST 000022    | Z.FLG 000010       |
| L\$MSK= 000037  | L.MPF 000022     | SF.ACT= 000000   | T\$NSTA 000005    | Z.LEN = 000016     |
| L\$RST= 000020  | L.NMST 000020    | SF.ENA= 000100   | T\$PDV 000007     | Z.LLN 000006       |
| L\$STP= 000010  | L.NSTA 000014    | SF.LPB= 000004   | T\$SLFG 000004    | Z.MAP 000020       |
| L\$STR= 000002  | L.OWNR 000021    | SF.MFL= 000040   | T\$SLF 000012     | Z.NAM 000004       |
| L\$CTL 000012   | L.UNT 000013     | SF.PAC= 000020   | T\$SLN 000010     | Z.PCB 000012       |
| L\$ERR 000010   | MAPAD 002354RG   | SF.REA= 000010   | T\$KMG= 000000    | Z.SCH 000007       |
| L\$FLG 000001   | MAXOV = ***** GX | SF.SER= 000001   | T\$MIN= 000000    | \$ACCESS= ***** GX |
| L\$FNC 000014   | M\$CRB= 000124   | SF.SVC= 000002   | T\$ST3 = ***** GX | \$BIAS = ***** GX  |
| L\$LEN 000026   | M\$CRX= 000000   | SF.UNL= 000040   | T3.PRIV= ***** GX | \$BYTES= ***** GX  |
| L\$LIN 000006   | M\$FCS= 000000   | SLTMA = ***** GX | V\$CTR= 001000    | \$DLXPD= ***** GX  |
| L\$LST 000000   | M\$MGE= 000000   | SLTNM = ***** GX | X\$DBT= 000000    | \$PBLK = ***** GX  |
| L\$LUN 000024   | M\$NET= 000000   | STACT 000020RG   | ZF.COU= 001000    | \$VA = ***** GX    |
| L\$RCV 000020   | M\$OVR= 000000   |                  | ZF.DDM= 000001    | .\$\$\$\$= 000034  |
| L\$RSR 000002   |                  |                  |                   |                    |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
002622 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 64  
Work file writes: 71  
Size of work file: 20638 words ( 81 Pages)  
Size of core pool: 17608 words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:35.11  
SY:DLXSUB.V2,[131,134]DLXSUB/CR/~SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]DLXSUB

```

83                                     .SBTTL EXEC entry point vector table
84
85 000012                               $EVLVT::
86                                     .IF DF R$$MPL
87                                     .WORD D.VINT ; 0 if vectored system, non-zero if not
88                                     .ENDC
89                                     .IF DF R$$MPL
90                                     .IF NDF R$$PRO
91 FMSK2: .WORD $FMSK2
92                                     .ENDC
93                                     .ENDC
94 000012 000000G TTNS: .WORD $TTNS
95 000014 000000G CEACC: .WORD $CEACC
96 000016 000000G EXRQN: .WORD $EXRQN
97 000020 000000G KSAR6: .WORD $KSAR6 ; for MAP, SAVMAP, RESMAP
98 000022 000000G LGDDB: .WORD $LGDDB
99 000024 000000G LGPDV: .WORD $LGPDV
100 000026 000000G LGSTT: .WORD $LGSTT
101
102                                     .IF DF R$$MPL
103                                     .IF NDF R$$PRO
104 MPLCK: .WORD $MPLCK
105                                     .ENDC
106                                     .ENDC
107
108 000030 000000G PDVID: .WORD $PDVID
109 000032 000000G PDVTA: .WORD $PDVTA
110 000034 000000G PSIPT: .WORD $PSIPT
111 000036 000000G SLTMA: .WORD $SLTMA
112 000040 000000G ZTIM2: .WORD $ZTIM2
113
114 000013                               $EVLVL==.-1-$EVLVT/2

```

|                 |                |                 |                |                    |
|-----------------|----------------|-----------------|----------------|--------------------|
| XF\$TIR= 000004 | X\$NNRE 000054 | X\$TIMC 000001  | ZF.MFL= 000010 | Z.SCH 000007       |
| XT\$CHN= 000100 | X\$NPL 000006  | X\$TIMR 000000  | ZF.MTM= 000400 | \$CEACC= ***** GX  |
| XT\$DGR= 000002 | X\$NPR 000007  | X\$TXI 000102   | ZF.MUX= 000040 | \$EVLTB 000000RG   |
| XT\$FAS= 000200 | X\$NRBY 000032 | X\$TXQ 000066   | ZF.PSE= 002000 | \$EVLVL= 000013 G  |
| XT\$INC= 000004 | X\$NRPK 000042 | X\$TYP 000015   | ZF.SLI= 010000 | \$EVLVI 000012RG   |
| XT\$OUC= 000010 | X\$NRRE 000053 | X\$USR 000022   | ZF.TIM= 000200 | \$EXRQN= ***** GX  |
| XT\$PVC= 000001 | X\$NTBY 000036 | X\$WAQ 000062   | ZF.X3P= 000000 | \$LGDB= ***** GX   |
| X\$ABQ 000076   | X\$NTPK 000046 | X\$WSZ 000020   | ZS.ASN= 100000 | \$LGLST= ***** GX  |
| X\$ALF 000055   | X\$PKSZ 000016 | X\$SDBT= 000000 | ZS.BSY= 140000 | \$LGPDV= ***** GX  |
| X\$AUC 000056   | X\$PR 000011   | ZERTAB 000422R  | ZTIM2 000040R  | \$LGSTT= ***** GX  |
| X\$CLEN= 000025 | X\$PRT 000021  | ZF.COU= 001000  | Z.AVL 000014   | \$LOST = ***** GX  |
| X\$CTIM 000060  | X\$PS 000010   | ZF.DDM= 000031  | Z.DAT 000016   | \$PDVID= ***** GX  |
| X\$DIAG 000003  | X\$RPR 000013  | ZF.DIA= 004000  | Z.DSP 000000   | \$PDVTA= ***** GX  |
| X\$DTE 000024   | X\$RPS 000012  | ZF.DLC= 000002  | Z.FLG 000010   | \$PSIPT= ***** GX  |
| X\$FLG 000014   | X\$RTRY 000002 | ZF.DVP= 100000  | Z.LEN = 000016 | \$SLTMA= ***** GX  |
| X\$GLEN 000106  | X\$RXI 000104  | ZF.INI= 040000  | Z.LLN 000006   | \$TTNS = ***** GX  |
| X\$LCN 000026   | X\$RXQ 000072  | ZF.KMX= 000020  | Z.MAP 000020   | \$ZTIM2= ***** GX  |
| X\$LEN 000106   | X\$SS 000004   | ZF.LLC= 000004  | Z.NAM 000004   | \$\$\$\$ = 000010  |
| X\$MOWN 000023  | X\$ST 000005   | ZF.LMC= 000100  | Z.PCB 000012   | .\$\$\$\$ = 000034 |
| X\$NLRE 000052  | X\$TCLZ 000030 | ZF.MAN= 020000  |                |                    |

. ABS. 177776 000 (RW,I,GBL,ABS,DVR)  
 001210 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 114  
 Work file writes: 108  
 Size of work file: 27608 Words ( 108 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:33.47  
 SY:EVL.V2,[131,134]EVL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]EVL

```

360 .SBTTL Move image mode field
361 +
362 *-MOVIMG-Move image mode field
363
364 Move an image mode field into the message buffer.
365
366 Inputs:
367 R0 = Pointer to data to be moved
368 R1 = Byte count of data to be moved
369 R2 = Pointer to next available byte in message buffer
370
371 Outputs:
372 R2 = Pointer past last byte used in message buffer
373
374 Registers modified:
375 R0, R1
376
377 000624 110122 MOVIMG: MOVB R1,(R2)+ ; Fill in byte count
378 000626 001403 BEQ 20$ ; If EQ, null field
379
380 000630 112022 10$: MOVB (R0)+,(R2)+ ; Copy image field
381 000632 SOB R1,10$ ;
382
383 000636 000241 20$: CLC
384 000640 RETURN

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                          |
|---------|------------|-------------------------------------|
| ACCLLT  | = ***** G  | 17-574 18-602                       |
| ADDLNK  | = ***** GX | 9-190                               |
| ADDMAI  | = ***** GX | 17-579                              |
| ADDOPT  | = ***** GX | 8-144 10-281                        |
| ASCBIN  | 000062 R   | 19-631 19-634 #20-665               |
| A.ACC   | 000020     | #6-57                               |
| A.LNK   | 000000     | #6-57                               |
| A.NAM   | 000002     | #6-57 14-444                        |
| A.PUD   | 000010     | #6-57                               |
| A.REM   | 000012     | #6-57 14-448                        |
| A.UCB   | 000010     | #6-57 14-439                        |
| BLDDSC  | 000532 R   | 10-257 10-261 #11-306               |
| BLDIMG  | 000610 R   | 10-266 10-271 10-276 #12-353        |
| BLDSES  | 000412 R   | 9-219 #10-253                       |
| CEACC   | = ***** GX | 14-435 14-487                       |
| CEMUL   | = ***** GX | 20-681                              |
| CHKACK  | = ***** GX | 18-604                              |
| CHKADD  | 000000 R   | 14-407 #19-625                      |
| CHKNAM  | 001256 R   | 14-412 14-417 14-445 14-493 #15-526 |
| CL\$MFL | = 000010   | 7-103                               |
| CL\$SFL | = 000004   | 7-106                               |
| CL\$TYP | = 000001   | 7-100                               |
| CL.MU1  | = 000001   | 10-264                              |
| CL.MU2  | = 000002   | 10-264                              |
| CMPINT  | 001414 RG  | #18-598                             |
| CTRSES  | = ***** GX | 9-222                               |
| CVTADD  | 000042 R   | 19-633 #19-637                      |
| CVTARE  | 000054 R   | 19-630 #19-643                      |
| CV\$40  | = 000002   | 8-140 9-215                         |
| C.FNC   | 000010     | *17-577                             |
| C.MOD   | 000011     | *17-578 *18-598                     |
| C.STS   | 000012     | 17-573 18-599                       |
| DECPT   | = ***** GX | 8-141 9-178 9-216 14-409            |
| D\$ANN  | 000000     | 14-433                              |
| D\$HOST | 000022     | 14-410                              |
| D\$LNUM | 000014     | 9-183 14-415 14-501                 |
| D\$RNN  | 000002     | 14-485                              |
| D\$SEC  | 000036     | 8-142 8-143 9-217 9-218             |
| ER\$FMT | = 000005   | 10-254                              |
| ER\$NOD | = 000002   | 14-404                              |
| E\$NBR  | 000014     | #6-49                               |
| E\$NBS  | 000020     | #6-49                               |
| E\$NCR  | 000034     | #6-49                               |
| E\$NCS  | 000036     | #6-49 9-222                         |
| E\$NIC  | 000044     | #6-49                               |
| E\$NLEN | 000050     | #6-49                               |
| E\$NLLA | 000012     | #6-49                               |
| E\$NLNK | 000000     | #6-49                               |
| E\$NML  | 000040     | #6-49                               |
| E\$NMR  | 000024     | #6-49                               |
| E\$NMS  | 000030     | #6-49                               |
| E\$NNOD | 000002     | #6-49                               |

```

95      .SBTTL Start or line assignment
96      ;+
97      ;**--START--Start or line assignment
98      ;
99      ; This routine is entered when another LLC (typically ourselves) has
100     ; requested that a line be started. The only action of this routine
101     ; is to set the current state enable for the line.
102     ;
103     ; Inputs:
104     ; R4 = Address of the control enable CCB
105     ;       C.NSP - Source PDV and Channel
106     ;       C.STS - State of the line (enable or disable) on a start
107     ; R5 = Address of the physical link access block
108     ;
109     ; Registers modified:
110     ; R0, R1, R2, R3, R4, R5
111     ;
112     000040 032764 000000G 000012 START: BIT    #ENBX,C.STS(R4) ; Should we change the state of enable?
113     000046 001415                      BEQ    10$          ; If EQ, no
114     ;
115     000050 042715 000400          BIC    #LF$ENB,(R5)      ; Assume state is disabled
116     000054 032764 000000G 000012 BIT    #LENB,C.STS(R4) ; Is the state disabled?
117     000062 001407                      BEQ    10$          ; If EQ, yes
118     ;
119     000064 052715 000400          BIS    #LF$ENB,(R5)      ; Mark the line as enabled
120     000070                      CALL   GTSLN              ; Get SLN and logical tributary #
121     000074                      CALL   STNMST              ; Get address of network management status byte
122     000100 105015                      CLRB   (R5)          ; Set state to on, closed
123     ;
124     000102 012700 000000G          10$:  MOV    #IS.SUC,R0      ; Set successful completion status
125     ;
126     ;+
127     ;**--CCBDN--Post completion on a control enable CCB
128     ;
129     ; Return the control enable CCB to the source process with a control
130     ; complete function code.
131     ;
132     ; Inputs:
133     ; R0 = Completion status
134     ; R4 = Address of the control enable CCB
135     ;       C.NSP - Source PDV and Channel
136     ;
137     000106 016464 000004 000006 CCBDN: MOV    C.NSP(R4),C.LIN(R4)
138     000114 112764 000020 000010      MOVB   #FC.CCP,C.FNC(R4)
139     000122 010064 000012          MOV    R0,C.STS(R4)      ; Set completion status
140     000126          CALLR @LLCRS      ; Post completion on the request
  
```

|                  |                |                  |                  |                  |
|------------------|----------------|------------------|------------------|------------------|
| ABORQ = ***** GX | CE.RTS= 100004 | CS.ENA= 000001   | C.XPT 000034     | FS.SPW= 040000   |
| ASYNCH 001012R   | CE.SRC= 100364 | CS.ENB= 000020   | C.XSVC 000042    | FS.STM= 000000   |
| A\$CHK= 000000   | CE.STP= 100352 | CS.ERR= 100000   | C.XTC 000037     | FS.STP= 002000   |
| A\$CPS= 000000   | CE.TME= 100354 | CS.FTL= 001000   | C.X25 000036     | FS.STR= 001000   |
| A\$PRI= 000000   | CE.TMO= 100374 | CS.HCR= 000001   | DISRQ = ***** GX | FS.TRM= 003000   |
| A\$TRP= 000000   | CE.UDF= 100001 | CS.HFE= 002000   | DQPKT = ***** GX | FS.WLB= 001000   |
| BLXIO = ***** GX | CE.UNS= 100344 | CS.IGN= 000002   | DSLIN = ***** GX | FS.XKL= 002000   |
| CB.CCB= 000002   | CF.CHN= 000001 | CS.LST= 040000   | D\$BUG= 177514   | FS.XOF= 010000   |
| CB.DDM= 000040   | CF.EDM= 000004 | CS.MTL= 004000   | D\$ISK= 000000   | FS.XUM= 007000   |
| CB.DLC= 000020   | CF.HDR= 000020 | CS.RNG= 000010   | D\$SLT1= 000001  | FS.ZER= 002000   |
| CB.RDB= 000004   | CF.LB = 100000 | CS.ROV= 000004   | D\$SYNC= 000000  | F\$SLVL= 000001  |
| CB.SDB= 000010   | CF.LIN= 000002 | CS.RSN= 010000   | D\$SYNM= 000000  | GTSLN = ***** GX |
| CB.SLI= 000100   | CF.SOM= 000010 | CS.SHU= 000001   | D.BIAS 000000    | G\$STPP= 000000  |
| CB.XLB= 000001   | CF.SYN= 000040 | CS.SID= 000002   | D.SIZE 000004    | G\$STSS= 000000  |
| CCBDN 000106R    | CF.TRN= 000100 | CS.STR= 000004   | ENBLC 001040R    | G\$STTK= 000000  |
| CCBRT = ***** GX | CHARC 001140R  | CS.SUC= 000001   | ENBX = ***** GX  | G\$SWRD= 000000  |
| CC.ADR= 000100   | CM.CIR= 000002 | CS.TMO= 020000   | E\$XPR= 000000   | IE.ABO= ***** GX |
| CC.DAD= 000102   | CM.DM1= 000253 | CS.XUR= 000004   | FC.CCP= 000020   | IE.DAD= ***** GX |
| CC.DPA= 000302   | CM.DM2= 000400 | CTLCCP= ***** GX | FC.CTL= 000006   | IE.TMO= ***** GX |
| CC.DST= 000200   | CM.DM3= 000000 | CTLCP 000706RG   | FC.KCP= 000016   | IE.VER= ***** GX |
| CC.ECM= 000311   | CM.FMT= 100000 | CTLDN = ***** GX | FC.KIL= 000004   | IO.XCH= 017400   |
| CC.ECS= 000310   | CM.HRD= 000002 | CTLDSP 000032R   | FC.MAN= 000024   | IO.XCL= 016210   |
| CC.ELS= 000304   | CM.LIN= 000000 | CTLEN 000022RG   | FC.MLD= 000026   | IO.XGC= 017410   |
| CC.EMC= 000300   | CM.LOD= 000001 | CTLTBL 000762R   | FC.PCT= 000030   | IO.XHG= 016000   |
| CC.EPH= 000303   | CM.LOI= 000317 | C\$CKP= 000000   | FC.PWR= 000022   | IO.XIN= 016400   |
| CC.EPR= 000301   | CM.LO2= 000000 | C\$ORE= 000400   | FC.RCE= 000002   | IO.XOP= 015400   |
| CC.LCR= 000004   | CM.LO3= 000000 | C\$SRSH= 177564  | FC.RCP= 000014   | IO.XRC= 015000   |
| CC.LDM= 000306   | CM.LT1= 000011 | C.ADD 000034     | FC.TIM= 000010   | IO.XSC= 017400   |
| CC.LLC= 000200   | CM.LT2= 000053 | C.BID 000003     | FC.XCP= 000012   | IO.XSP= 017010   |
| CC.LSA= 000307   | CM.LT3= 007400 | C.BUF 000014     | FC.XME= 000000   | IO.XST= 017000   |
| CC.MCT= 000201   | CM.RC1= 000253 | C.BUF1 000014    | FNCAB = ***** GX | IO.XTL= 017000   |
| CC.NET= 000000   | CM.RC2= 001000 | C.BUF2 000024    | FS.AST= 000000   | IO.XTM= 014400   |
| CC.PDC= 000006   | CM.RC3= 000000 | C.CHRL 000010    | FS.CIB= 002000   | IO.XTR= 016000   |
| CC.PRO= 000101   | CM.XLO= 000004 | C.CNT 000020     | FS.CRA= 001000   | IS.SUC= ***** GX |
| CC.SAD= 000010   | CP.CON= 001140 | C.CNT1 000020    | FS.DIS= 013000   | ITRANS= ***** GX |
| CC.SID= 000305   | CP.DCF= 000040 | C.CNT2 000030    | FS.DVC= 001000   | I\$RAR= 000000   |
| CC.SPL= 000002   | CP.DUM= 000540 | C.DAT1 000002    | FS.ENB= 012000   | I\$RDN= 000000   |
| CC.STM= 000312   | CP.HDL= 000007 | C.DAT2 000004    | FS.EXI= 001000   | I.FCN = ***** GX |
| CE.ABO= 100362   | CP.LAT= 002140 | C.FLG 000022     | FS.GET= 006000   | I.LN2 = ***** GX |
| CE.ACN= 100012   | CP.LOO= 000220 | C.FLG1 000025    | FS.HLT= 000000   | I.PRM = ***** GX |
| CE.DAD= 100346   | CP.PS = 177400 | C.FLG2 000035    | FS.INJ= 000000   | I.UCB = ***** GX |
| CE.DIS= 100366   | CP.PSI= 000200 | C.FNC 000010     | FS.KIL= 000000   | KISAR6= ***** GX |
| CE.DNF= 100005   | CP.ROU= 001540 | C.LIN 000006     | FS.LCL= 100000   | K\$CNT= 177546   |
| CE.ERR= 100370   | CP.XCF= 000100 | C.LNK 000000     | FS.LTM= 001000   | K\$CSR= 177546   |
| CE.ILC= 100006   | CP.2FR= 000030 | C.MOD 000011     | FS.MNT= 004000   | K\$SLDC= 000000  |
| CE.ILN= 100350   | CRCOV= 000000  | C.NSP 000004     | FS.MSN= 014000   | K\$STPS= 000074  |
| CE.INV= 100002   | CS.ABO= 000100 | C.PRO 000042     | FS.REA= 001000   | LDLPL = 000000   |
| CE.IUM= 100013   | CS.BRO= 000002 | C.RSV 000002     | FS.RET= 000000   | LENB = ***** GX  |
| CE.LTO= 100356   | CS.BUF= 000200 | C.STA 000007     | FS.REZ= 003000   | LF\$BRD= 002000  |
| CE.MCE= 100007   | CS.CES= 000002 | C.STAT 000006    | FS.RLB= 002000   | LF\$CHN= 004000  |
| CE.MOP= 100372   | CS.CHN= 000010 | C.STS 000012     | FS.RNG= 011000   | LF\$ENB= 000400  |
| CE.NMA= 100014   | CS.CMP= 000200 | C.TYP 000000     | FS.RST= 000000   | LF\$MOP= 001000  |
| CE.NTE= 100361   | CS.DAO= 000003 | C.URM 177776     | FS.RTN= 001000   | LF\$RSR= 100000  |
| CE.PCN= 100011   | CS.DCR= 000400 | C.XACP 000004    | FS.SET= 005000   | LF.ACT= 100000   |
| CE.RES= 100010   | CS.DEF= 000004 | C.XID 000035     | FS.SFC= 005000   | LF.BRO= 000400   |
| CE.RTE= 100376   | CS.DEV= 000002 | C.XLEN 000044    | FS.SFR= 006000   | LF.BWT= 000007   |
| CE.RTL= 100003   | CS.DIS= 000040 | C.XPLI 000040    | FS.SFS= 004000   | LF.ENA= 002000   |



```

141 .SBTTL Retrieve Ethernet protocol
142
143 ;+
144 **--RCVPRO-Retrieve Ethernet protocol
145 Retrieve the Ethernet protocol for this received message.
146
147 Inputs:
148 R0 = Address of characteristics block
149 R1 = Default status (CS.IGN = Characteristic ignored)
150 R3 = Address of the I/O packet
151 R4 = Address of the receive CCB
152 C.BUF - Bias of received message
153 C.BUF+2 - Virtual address of received message
154 C.CNT - Length of received message
155 C.NSP - Virtual address of Ethernet header
156 C.PRO - Protocol type from message
157 C.ADD - Source address from Ethernet message
158 R5 = Address of the physical link access block
159
160 Outputs:
161 R1 = Status to return for characteristic
162
163 000146 026027 000002 000002 RCVPRO: CMP C.DATI(R0),#2 ; Are there at least 2 bytes available?
164 000154 103413 10$ ; If L0, no
165
166 000156 032715 002000 BIT #LF$BRO,(R5) ; Is this a broadcast channel?
167 000162 001410 BEQ 10$ ; If EQ, no
168
169 000164 016460 000042 000010 MOV C.PRO(R4),C.CHRL(R0)
170 000172 012760 000002 000004 MOV #2,C.DATO(R0) ; Fill in Ethernet protocol and it's size
171 000200 012701 000001 MOV #CS.SUC,R1 ; Return successful completion
172
173 000204 10$: RETURN

```

DLXCTL - DLX control Q10 functi MACRO V05.03b Friday 28-Jun-85 22:54<sup>L 6</sup> Page 6  
Define local macros

54  
55  
56  
57  
58  
59

.SBTTL Define local macros  
.MACRO DISP,TYPE,ADDR  
NX.'TYPE==,-CCPTBL  
.WORD ADDR  
.ENDM DISP

```

449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467 000710 116764 000000G 000005 QCCBL:: MOVB $DLXPD,C.NSP+1(R4)
468 000716 010064 000014 MOV R0,C.BUF(R4) ; Save target PDV and Channel
469 000722 010164 000016 MOV R1,C.BUF+2(R4) ; Save address in PDV channel table
470 000726 010264 000020 MOV R2,C.CNT(R4) ; Save SLN and logical tributary #
471 000732 010364 000022 MOV R3,C.FLG(R4) ; Save I/O packet address
472 000736 CALLR @LLCRS ; Queue request to the LLC

.SBTTL Queue a control CCB to the target LLC
+
**QCCBL-Queue a control CCB to the target LLC
Queue a control CCB to the target LLC to perform some line control
function.
-
Inputs:
R0 = Target PDV and Channel
R1 = Address in the PDV channel table
R2 = SLN and logical tributary #
R3 = Address of I/O packet
R4 = Address of the control CCB (with the following fields set up):
C.NSP - Next state index
C.LIN - Target PDV and Channel
C.FNC - Function code for target LLC
C.STS - Additional control for target

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

```
.TITLE DLXLIN - DLX Line control routines
.IDENT /V05.00/
.ENABL LC
```

```
: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.
```

```
: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.
```

```
: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.
```

```
: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.
```

```
: Module description
```

```
: DLX Line control routines
```

```
: Distributed Systems Software Engineering
```

```
: Ident History:
```

```
: 4.00 07-NOV-83
: DECnet-11M V4.0
: DECnet-11M-PLUS V2.0
```

```
: 5.00 22-JUL-85
: DECnet-11M/S V4.2
: DECnet-11M-Plus V3.0
: DECnet-Micro/RSX V1.0
```

DLXQIO - ISSUE I/O TO DLX      MACRO V05.03b Saturday 29-Jun-85 L 10  
Table of contents      12:23

|    |     |                                   |
|----|-----|-----------------------------------|
| 5- | 55  | MACRO CALLS AND LOCAL DATA        |
| 6- | 70  | CONTEXT AREA OFFSET DEFINITIONS   |
| 7- | 78  | ERROR MESSAGE STRINGS             |
| 8- | 91  | DLXQIO - ISSUE I/O REQUEST TO NX: |
| 9- | 186 | DLXLUN - ASSIGN A LUN TO NX:      |

DLXQIO      CREATED BY    MACRO    ON 29-JUN-85 AT 12:23      PAGE 1      L 11  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES           |
|---------|------------|----------------------|
| DEALUN  | = ***** GX | 9-202                |
| DLXERR  | 000020 R   | #7-88 8-176          |
| DLXLUN  | 000200 RG  | #9-200               |
| DLXMSG  | 000004 R   | #7-86                |
| DLXQIO  | 000026 RG  | #8-110               |
| DLXTXT  | 000005 R   | #7-87 8-179          |
| FMTLIN  | = ***** GX | 8-138                |
| G.LUNA  | = 000000   | 9-206                |
| IE.UPN  | = ***** GX | 8-158                |
| IOSB    | 000000 R   | #5-66 8-149 8-165    |
| LC\$NTL | = 000200   | #6-75                |
| LC\$OWN | = 000400   | #6-75                |
| LF\$MLT | = 040000   | #6-75                |
| LF\$REA | = 000001   | #6-75                |
| LF\$SEG | = 100000   | #6-75                |
| LF\$SKP | = 000004   | #6-75                |
| LF\$VR2 | = 000010   | #6-75                |
| LF\$ZER | = 000002   | #6-75                |
| LP\$MPT | = 000010   | #6-75                |
| LP\$MUX | = 000004   | #6-75                |
| LP\$NXC | = 100000   | #6-75                |
| LP\$TRB | = 000002   | #6-75                |
| LP\$UNT | = 000001   | #6-75                |
| LP\$WCN | = 000040   | #6-75 6-75           |
| LP\$WDV | = 000020   | #6-75 6-75           |
| LP\$WLD | = 000360   | #6-75                |
| LP\$WTR | = 000200   | #6-75 6-75           |
| LP\$WUN | = 000100   | #6-75 6-75           |
| LS\$ACT | = 000040   | #6-75                |
| LS\$BPT | = 010000   | #6-75                |
| LS\$E.. | = 001000   | #6-75 6-75           |
| LS\$E.. | = 000001   | #6-75                |
| LS\$E.. | = 000002   | #6-75                |
| LS\$BSI | = 000004   | #6-75                |
| LS\$CHN | = 000001   | #6-75 6-75           |
| LS\$CMB | = 000002   | #6-75 6-75 6-75 6-75 |
| LS\$COS | = 000004   | #6-75                |
| LS\$CUS | = 000004   | #6-75 6-75           |
| LS\$DDT | = 001000   | #6-75                |
| LS\$DEA | = 000100   | #6-75                |
| LS\$DLM | = 001004   | #6-75                |
| LS\$DLT | = 002000   | #6-75                |
| LS\$DTE | = 000010   | #6-75 6-75           |
| LS\$HBT | = 000400   | #6-75                |
| LS\$HTM | = 010000   | #6-75                |
| LS\$INA | = 000010   | #6-75                |
| LS\$IND | = 000020   | #6-75                |
| LS\$INI | = 000040   | #6-75                |
| LS\$LCT | = 020000   | #6-75 6-75           |
| LS\$LMB | = 000002   | #6-75                |
| LS\$LOO | = 040000   | #6-75                |
| LS\$MDT | = 020000   | #6-75                |

```
227 .SBTTL Convert decimal to binary
228 *
229 **--CDTB-Convert decimal to binary
230
231 Convert a string of ASCII decimal characters to binary.
232
233 --
234 Inputs:
235 R0 = Address of first byte in the <Line-id> string
236 R3 = Legal terminating character
237 R4 = Address of the end of the string +1
238
239 Outputs:
240 R0 = Address of next byte in the <Line-id> string
241 R1 = Converted result
242 'C' Clear - Successful conversion
243 'C' Set - Illegal character in the string
244
245 CDTB:: SAVRG <R2> ; Get a free register
246 CLR R1 ; Clear accumulation
247
248 10$: MOVB (R0)+,R2 ; Get the next character
249 CMP R4,R0 ; Reached the end of the string yet?
250 BLO 20$ ; If LO, yes
251 CMPB R3,R2 ; Is this a legal terminating character?
252 BEQ 20$ ; If EQ, yes
253
254 SUB #10,R2 ; Normalise the character
255 CMP R2,#10. ; Is it within range?
256 BHS 30$ ; If HS, no
257
258 SAVRG <R0> ; Save register
259 MOV #10.,R0 ; Set up the multiplicand
260 CALL @CEMUL ; Multiply it out
261 RESRG <R0> ; Restore register
262 ADD R2,R1 ; Accumulate result
263 BR 10$ ; and loop
264
265 20$: BIT #177400,R1 ; Is the result a byte quantity?
266 BNE 30$ ; If NE, no
267 TST (PC)+ ; Indicate success
268 SEC ; Illegal character or number too large
269 RESRG <R2> ; Restore register
270 RETURN
```

|     |        |        |        |
|-----|--------|--------|--------|
| 244 | 000346 |        |        |
| 245 | 000350 | 005001 |        |
| 246 |        |        |        |
| 247 | 000352 | 112002 |        |
| 248 | 000354 | 020400 |        |
| 249 | 000356 | 103417 |        |
| 250 | 000360 | 120302 |        |
| 251 | 000362 | 001415 |        |
| 252 |        |        |        |
| 253 | 000364 | 162702 | 000060 |
| 254 | 000370 | 020227 | 000012 |
| 255 | 000374 | 103014 |        |
| 256 |        |        |        |
| 257 | 000376 |        |        |
| 258 | 000400 | 012700 | 000012 |
| 259 | 000404 |        |        |
| 260 | 000410 |        |        |
| 261 | 000412 | 060201 |        |
| 262 | 000414 | 000756 |        |
| 263 |        |        |        |
| 264 | 000416 | 032701 | 177400 |
| 265 | 000422 | 001001 |        |
| 266 | 000424 | 005727 |        |
| 267 | 000426 | 000261 |        |
| 268 | 000430 |        |        |
| 269 | 000432 |        |        |



DLXSUB - DLX subroutines  
Scan databases for <Line-id>

MACRO V05.03b Friday 28-Jun-85 22:56 Page 18-1

L 13

```

758 001670 116167 000014 000005G      MOVB    L,NSTA(R1), $PBLK+T$NSTA
759 001676 001422      BEQ     50$      ; If EQ, can't be a multi-point master
760 001700 032711      BIT     #LF.MTP, (R1) ; Is this a multi-point line?
761 001704 001417      BEQ     50$      ; If EQ, no
762 001706 116700      MOVVB   $PBLK+T$SLN+1,R0; Get logical tributary #
763 001712 120061 000014      CMPEB   R0,L,NSTA(R1) ; Is it valid for this line?
764 001716 103342      BHIS    30$      ; If HIS, no
765
766 001720 006300      ASL      R0      ; Form double word offset
767 001722 006300      ASL      R0      ;
768 001724 060100      ADD      R1,R0      ; Offset into system line table
769 001726 116067      MOVVB   L.MPF(R0), $PBLK+T$SFLG
770 001734 016067 000022 000004G      MOV     L.MPF+S.NMST(R0), $PBLK+T$NMST
771 001742 000414      BR       60$      ; Re-enter common code
772
773 001744 105767 000011G      50$:   TSTB    $PBLK+T$SLN+1 ; Did user specify a tributary #?
774 001750 001325      BNE      30$      ; If NE, yes
775 001752 032767 000400 000012G      BIT     #LF.BRO, $PBLK+T$SLF ; Is this a broadcast channel?
776 001760 001405      BEQ     60$      ; If EQ, no
777 001762 005767 000000G      TST     $ACCESS ; Should we dynamically assign a port?
778 001766 001002      BNE      60$      ; If NE, no
779 001770      CALLR   FNDDBR ; Find a free broadcast port
780
781 001774 006302      60$:   ASL      R2      ; Compute address of entry
782 001776 067702      ADD      @LLCTA,R2 ; in the reverse mapping table
783 002002 011200      MOV      (R2),R0 ; and get the contents
784 002004 100004      BPL      70$      ; If PL, point-to-point connection
785 002006 116702 000011G      MOVVB   $PBLK+T$SLN+1,R2; Get the logical tributary #
786 002012 060002      ADD      R0,R2      ; Compute address of entry
787 002014 006302      ASL      R2      ; in the reverse mapping table
788
789 002016 011267 000016G      70$:   MOV      (R2), $PBLK+T$ASN; Store current PDV and Channel
790
791 002022 000241      90$:   CLC      ; Indicate success
792 002024      100$:  RETURN

```

M 13

DLXSUB      CREATED BY    MACRO    ON 28-JUN-85 AT 22:56      PAGE 1      L 14

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|---|
| ACHCK   | = ***** GX  | 7-155   |
| ASSIGN  | = 000000 RG | #6-69   |
| BLXIO   | = ***** GX  | 14-472  |
| BUFCHK  | 000104 RG   | #7-140  |
| CAT5    | 000224 RG   | #8-186      19-836  |
| CDTB    | 000346 RG   | #9-244      19-853      19-863      19-870  |
| CEMUL   | = ***** GX  | 8-203      9-259      25-1035   |
| CHKCHR  | 000434 RG   | 7-162      #10-285  |
| CLACT   | 000610 RG   | #12-374   |
| COASN   | 000526 RG   | 6-69      #11-325      12-358   |
| CVTBL   | 000340 R    | 8-188      8-207      #8-223  |
| C.BUF   | 000014      | *14-451      14-470      14-471   |
| C.CHRL  | 000010      | 10-293      10-300      10-302  |
| C.CNT   | 000020      | *14-452      14-459      *14-463  |
| C.DATI  | 000002      | 10-295      10-299      10-301  |
| C.LIN   | 000006      | 25-1029   |
| C.STAT  | 000006      | *10-299   |
| DEASSN  | 000576 RG   | #12-374      13-415   |
| DSLIN   | 000716 RG   | #13-414   |
| FILCHN  | 000726 RG   | #14-439   |
| FINDC   | 001246 RG   | #16-567   |
| FINDL   | 001262 RG   | #16-599   |
| FNDBR   | 001142 RG   | #15-508      18-779   |
| FNDFR   | 001350 RG   | #17-660   |
| FNDLA   | 001504 RG   | #18-718      19-876   |
| FNDLN   | 002026 RG   | 16-600      #19-816   |
| FNDPR   | 002254 RG   | #20-903   |
| GTSLN   | 002320 RG   | 13-414      #21-931   |
| IE.MOD  | = ***** GX  | 23-992  |
| IE.MOD  | = ***** GX  | 17-689  |
| IE.NSF  | = ***** GX  | 16-635  |
| IE.PRI  | = ***** GX  | 16-632  |
| IE.RSU  | = ***** GX  | 17-661  |
| IE.SPC  | = ***** GX  | 7-165   |
| ISSAS   | = *****     | 5-48  |
| I.PRM   | = ***** GX  | 7-140      *7-143      *7-144      *7-145      7-148      *7-153      *7-154      *7-158      *7-159  |
|         |             | *7-160      7-161      14-441      14-442      14-443      19-829      19-831      19-832      20-903 |
| I.TCB   | = ***** GX  | 16-567  |
| KISAR6  | = ***** GX  | *10-287      *19-829  |
| LDBGT   | = ***** GX  | 14-447  |
| LF\$RSR | = 100000    | 14-480  |
| LF.ACT  | = 100000    | #5-50      6-93      12-396   |
| LF.BRO  | = 000400    | #5-50      18-775      26-1058  |
| LF.BWT  | = 000007    | #5-50   |
| LF.ENA  | = 002000    | #5-50   |
| LF.LPB  | = 001000    | #5-50      17-667   |
| LF.MDC  | = 000100    | #5-50   |
| LF.MFL  | = 004000    | #5-50   |
| LF.MTP  | = 000020    | #5-50      17-662      18-760      26-1060  |
| LF.PAC  | = 000200    | #5-50   |
| LF.RDY  | = 040000    | #5-50      16-605   |
| LF.REA  | = 010000    | #5-50   |

```
116 .SBTTL LOG NETWORK EVENT
117
118 *
119 ***NELOG-LOG NETWORK EVENT
120 LOG NETWORK EVENT FROM A COMM/EXEC PROCESS. IF THE EVENT IS PRESENT
121 IN THE COUNTER CONTROL TABLE, THE INCOMING BUFFER AS DESCRIBED BY
122 THE EVENT CONTROL MASK AND R4 WILL BE ASSUMED TO BE A COUNTER BLOCK.
123 THE ENTIRE COUNTER BLOCK WILL BE ZEROED AND THE FIRST WORD WILL BE
124 FILLED IN WITH THE TIME SINCE MIDNIGHT.
125
126 -
127 INPUTS:
128 R0 - EVENT CLASS AND TYPE
129 R1 - POINTER TO 3 WORD CONTROL BLOCK:
130 .WORD EVENT CONTROL MASK
131 .WORD POINTER TO EVENT DESCRIPTOR BLOCK
132 .WORD CALLING PDV INDEX
133
134 R4 - CCB ADDRESS, DATA AREA POINTER OR LINE-ID
135 R5 - POINTER TO EVENT LOGGER PROCESS DATABASE
136
137 OUTPUTS:
138 'C' CLEAR - EVENT SUCCESSFULLY PROCESSED
139 'C' SET - ERROR PROCESSING EVENT
140
141 REGISTERS MODIFIED:
142 R0, R1, R2, R3, R4
143
144 NELOG: TST E$TCB(R5) ; IS THE EVENT LOGGER RUNNING?
145 BNE 5$ ; BR IF YES
146 JMP 110$ ; IF EQ, NO ... CAN'T LOG THE EVENT
147 5$: CALL ALLOC ; ALLOCATE AN EVENT BUFFER
148 BCS 65$ ; IF CS, ALLOCATION FAILURE
149
150 SAVRG <R0,R1,R4> ; SAVE INITIAL CALLING PARAMETERS
151 MOV R0,E$EVT(R3) ; FILL IN EVENT CLASS AND TYPE
152 MOV R3,R0 ; COPY ADDRESS OF EVENT BUFFER
153 ADD #E$CTL,R0 ; POINT TO CONTROL MASK
154 MOVB (R1),(R0)+ ; FILL IN CONTROL MASK
155 MOVB 4(R1),(R0)+ ; AND CALLING PDV INDEX
156 CLR (R0)+ ; CLEAR OUT SIZE OF ADDITIONAL DATA
157 MOV 2(R1),R2 ; GET POINTER TO EVENT DESCRIPTOR BLOCK
158
159 MOV (R2)+,(R0)+ ; FILL IN LINE-ID
160
161 .REPT 3
162 MOV (R2)+,(R0)+ ; FILL IN EVENT PARAMETERS
163 .ENDR
164
165 MOV (R2)+,(R0)+ ; FILL IN REMOTE NODE ADDRESS
166 MOV (R2)+,(R0)+ ; FILL IN MODULE ID
167 MOV (R2)+,(R0)+ ; FILL IN X.25 PORT #
168 MOV (R2)+,(R0)+ ; FILL IN X.25 LOGICAL CHANNEL #
169
170 BITB #EV.CCB,E$CTL(R3)
171 BEQ 20$ ; IF EQ, NO CCB PRESENT
172 BITB #EV.LCB,E$CTL(R3)
173 BEQ 10$ ; IF EQ, USE LINE-ID FROM EVENT DESCRIPTOR BLOCK
```

|     |        |        |               |
|-----|--------|--------|---------------|
| 143 | 000042 | 005765 | 000004        |
| 144 | 000046 | 001002 |               |
| 145 | 000050 | 000167 | 000344        |
| 146 | 000054 |        |               |
| 147 | 000060 | 103525 |               |
| 149 | 000062 |        |               |
| 150 | 000070 | 010063 | 000002        |
| 151 | 000074 | 010300 |               |
| 152 | 000076 | 062700 | 000020        |
| 153 | 000102 | 111120 |               |
| 154 | 000104 | 116120 | 000004        |
| 155 | 000110 | 005020 |               |
| 156 | 000112 | 016102 | 000002        |
| 158 | 000116 | 012220 |               |
| 160 |        | 000003 |               |
| 164 | 000126 | 012220 |               |
| 165 | 000130 | 012220 |               |
| 166 | 000132 | 012220 |               |
| 167 | 000134 | 012220 |               |
| 169 | 000136 | 132763 | 000001 000020 |
| 170 | 000144 | 001414 |               |
| 171 | 000146 | 132763 | 000100 000020 |
| 172 | 000154 | 001403 |               |

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES |         |        |        |      |      |      |      |
|---------|------------|------------|---------|--------|--------|------|------|------|------|
| AC\$DNT | = 000002   | #5-61      |         |        |        |      |      |      |      |
| AC\$X25 | = 000001   | #5-61      |         |        |        |      |      |      |      |
| AE\$CIR | = 000003   | #5-61      |         |        |        |      |      |      |      |
| AE\$LIN | = 000001   | #5-61      |         |        |        |      |      |      |      |
| AE\$MOD | = 000004   | #5-61      |         |        |        |      |      |      |      |
| ALOC    | = 000532 R | 9-146      | #11-335 |        |        |      |      |      |      |
| CEACC   | = 000014 R | #8-95      | 13-427  | 13-433 | 13-448 |      |      |      |      |
| CL\$ASZ | = 010500   | #5-61      |         |        |        |      |      |      |      |
| CL\$DLL | = 000500   | #5-61      | 5-61    | 5-61   | 5-61   | 5-61 | 5-61 | 5-61 | 5-61 |
| CL\$ECL | = 000300   | #5-61      | 5-61    |        |        |      |      |      |      |
| CL\$LDN | = 010400   | #5-61      | 5-61    |        |        |      |      |      |      |
| CL\$MAN | = 000000   | #5-61      | 5-61    | 5-61   |        |      |      |      |      |
| CL\$PAZ | = 034100   | #5-61      | 5-61    | 5-61   |        |      |      |      |      |
| CL\$PLH | = 034000   | #5-61      | 5-61    | 5-61   | 5-61   |      |      |      |      |
| CL\$PLL | = 000600   | #5-61      |         |        |        |      |      |      |      |
| CL\$PRT | = 034200   | #5-61      |         |        |        |      |      |      |      |
| CL\$ROU | = 010000   | #5-61      | 5-61    | 5-61   |        |      |      |      |      |
| CL\$SES | = 000200   | #5-61      | 5-61    | 5-61   |        |      |      |      |      |
| CL\$SGE | = 035000   | #5-61      | 5-61    | 5-61   |        |      |      |      |      |
| CL\$SSE | = 035100   | #5-61      | 5-61    | 5-61   | 5-61   | 5-61 | 5-61 | 5-61 | 5-61 |
| CL\$TRN | = 000400   | #5-61      | 5-61    | 5-61   | 5-61   | 5-61 | 5-61 | 5-61 | 5-61 |
| CL\$XL2 | = 013700   | #5-61      |         |        |        |      |      |      |      |
| CL\$XL3 | = 013600   | #5-61      | 5-61    |        |        |      |      |      |      |
| CL\$X2S | = 013500   | #5-61      | 5-61    |        |        |      |      |      |      |
| C\$PORT | = 000014   | 13-451     |         |        |        |      |      |      |      |
| C\$BUF  | = 000014   | 9-174      | 9-175   | 9-255  |        |      |      |      |      |
| C\$LIN  | = 000006   | 9-173      |         |        |        |      |      |      |      |
| DL\$AST | = 000002   | #5-61      |         |        |        |      |      |      |      |
| DL\$HLT | = 000000   | #5-61      |         |        |        |      |      |      |      |
| DL\$IST | = 000001   | #5-61      |         |        |        |      |      |      |      |
| DL\$MAI | = 000004   | #5-61      |         |        |        |      |      |      |      |
| DL\$OFF | = 000001   | #5-61      |         |        |        |      |      |      |      |
| DL\$ON  | = 000000   | #5-61      |         |        |        |      |      |      |      |
| DL\$RUN | = 000003   | #5-61      |         |        |        |      |      |      |      |
| DL\$SHU | = 000002   | #5-61      |         |        |        |      |      |      |      |
| DL\$SYN | = 000005   | #5-61      |         |        |        |      |      |      |      |
| EV\$ACT | = 000001   | #5-61      |         |        |        |      |      |      |      |
| EV\$ACF | = 000201   | #5-61      |         |        |        |      |      |      |      |
| EV\$ADR | = 000420   | #5-61      |         |        |        |      |      |      |      |
| EV\$ADU | = 000417   | #5-61      |         |        |        |      |      |      |      |
| EV\$APL | = 000400   | #5-61      |         |        |        |      |      |      |      |
| EV\$ARC | = 000421   | #5-61      |         |        |        |      |      |      |      |
| EV\$AUC | = 000010   | #5-61      |         |        |        |      |      |      |      |
| EV\$AUS | = 000003   | #5-61      |         |        |        |      |      |      |      |
| EV\$CDF | = 000520   | #5-61      |         |        |        |      |      |      |      |
| EV\$COZ | = 000011   | #5-61      |         |        |        |      |      |      |      |
| EV\$DBR | = 000302   | #5-61      |         |        |        |      |      |      |      |
| EV\$GAS | = 035101   | #5-61      |         |        |        |      |      |      |      |
| EV\$HCE | = 035114   | #5-61      |         |        |        |      |      |      |      |

```

386 .SBTTL Map node name to node address
387
388 ***-MAPNAM-Map node name to node address
389
390 Scan the alias and remote name lists to provide a mapping from the
391 node name to the node address.
392
393 -
394 Inputs:
395 R5 = Address of database descriptor
396
397 Outputs:
398 'C' Clear - Valid node name mapping found
399 'C' Set - Unable to find a node name mapping
400 $FLAGS - bit 0 is set if the node is loopback node
401
402 Registers modified:
403 R0, R2, R3, R4
404
405 000642 012705 000002 000022 MAPNAM: MOV #ERSNOD,N$ERRC(R5)
406 000650 142767 000001 000000G BICB #1,$FLAGS ; Initialize as not a loopback node
407 000656 RECMAP ; Recover APR6 mapping
408 000664 CALL CHKADD ; Is node name an ascii node address?
409 000670 103155 BCC 97$ ; If cc, yes
410 000672 017704 000000G MOV @DECPT,R4 ; Point to the DEC home block
411 000676 016465 000022 000012 MOV D$HOST(R4),N$SNOD(R5) ; Assume name is 'HOST'
412 000704 012700 000000G MOV #HOST,R0 ; Check if it really is
413 000710 CALL CHKNAM ; Check for name match
414 000714 001555 BEQ 100$ ; If EQ, it is
415
416 000716 016465 000014 000012 MOV D$LNUM(R4),N$SNOD(R5) ; Assume name is 'LOCAL'
417 000724 012700 000000G MOV #LOCAL,R0 ; Check if it really is
418 000730 CALL CHKNAM ; Check for name match
419 000734 001545 BEQ 100$ ; If EQ, it is
420
421 .IF DF N$SLI
422
423 MOV #-1,R3 ; Force no TI: match for SLI interface
424 BITB #LT.SLI,$LTYPE ; Is this a system interface link?
425 BNE 5$ ; If NE, yes
426
427 .ENDC
428
429 000736 016703 000000G MOV $IOPKT,R3 ; Recover I/O packet address
430 000742 016303 000000G MOV 17CB(R3),R3 ; Find requesting task's TI:
431 000746 016303 000000G MOV TUCB(R3),R3 ; ...
432
433 000752 010402 000000 5$: MOV R4,R2 ; Scan the alias list first
434 000754 062702 000000 ADD #D$ANN,R2 ; Point to the alias listhead
435 000760 011246 10$: MOV (R2),-(SP) ; Get address of next entry
436 000766 012602 MOV @CEACC ; Gain access to the block
437 000770 001461 MOV (SP)+,R2 ; Retrieve virtual address
438 BEQ 80$ ; If EQ, no more entries
439
440 000772 016200 000010 MOV AUCB(R2),R0 ; Get UCB for this entry
441 000776 001402 BEQ 20$ ; If EQ, global alias
442 001000 020003 CMP R0,R3 ; Is entry for this task's TI?:
443 001002 001366 BNE 10$ ; If NE, no ... keep looking

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|---|
| ESNRT   | 000042     | #6-49   |
| ESNRT   | 000005     | #6-49   |
| ESNSEG  | 000010     | #6-49   |
| ESNTIM  | 000046     | #6-49   |
| ESNUSE  | 000004     | #6-49   |
| ESSTRT  | 000006     | #6-49   |
| FLSRES  | = ***** GX | 18-600  |
| GETLDB  | = ***** GX | 9-209   |
| GFTSDB  | = ***** GX | 8-134   |
| HOST    | = ***** GX | 14-411  |
| J.TCB   | = ***** GX | 14-429  |
| KISAR6  | = ***** GX | *8-129 *14-406  |
| LDBRT   | = ***** GX | 9-227   |
| LF.MMF  | = 000200   | 7-101   |
| LF.MSF  | = 000100   | 7-104   |
| LOCAL   | = ***** GX | 14-416  |
| LT.LCL  | = 000001   | 9-188   |
| LT.LPL  | = 000002   | 9-182   |
| L.DCR   | 000100     | *16-549   |
| L.FLAG  | 000014     | *7-96 7-101 7-104   |
| L.ILTT  | 000066     | *7-78   |
| L.LSF1  | 000044     | *18-603   |
| L.REM   | 000006     | *9-201  |
| L.USTA  | 000036     | *16-552 *16-555   |
| L.WIND  | 000040     | 17-575  |
| MAPNAM  | 000642 RG  | 9-173 #14-404   |
| MC.CC   | = 000040   | 8-136   |
| MC.CI   | = 000020   | 9-211   |
| MF.CTL  | = 000010   | 8-136 9-211   |
| MOVIMG  | 000624 R   | 11-326 12-356 #13-377   |
| NC.FM2  | = 000002   | 11-311  |
| NMSARA  | = 176000   | 14-499 14-502   |
| NS\$SDI | = 000002   | 16-552  |
| NT\$CC  | = 000016   | 8-135   |
| NT\$CON | = 000000   | 9-210   |
| NT\$INT | = 000004   | 18-598  |
| NT.INT  | = 000002   | 17-577  |
| NSERRC  | 000022     | *10-254 *14-404   |
| NSLLTM  | 000024     | 8-129   |
| NS\$LA  | 000016     | *9-176  |
| NS\$NOD | 000012     | 9-179 *9-183 *14-410 *14-415 *14-498 14-499 *14-503 *19-629 *20-685 |
| NS\$ECL | = *****    | 9-222   |
| NS\$EVL | = 000001   | #4-2  |
| NS\$MLL | = 000001   | 8-129   |
| NS\$NCT | = 000001   | 9-222   |
| NS\$SES | = 000001   | #6-62   |
| NS\$SLI | = *****    | 7-80 14-420   |
| NS\$VCT | = *****    | 8-129 8-131 9-203 9-223 14-406 17-574 18-600 18-602                 |
| REMLNK  | = ***** GX | 18-604  |
| RF.LOO  | = 100000   | 8-150 9-228   |
| RTRSES  | = ***** GX | #6-56 14-495  |
|         |            | 8-131 9-203   |

```

142 .SBTTL Line shutdown
143
144 **--SHUT-Line shutdown
145
146 This routine is entered when another LLC (typically ourselves) has
147 requested that a line be shutdown.
148
149 Inputs:
150 R4 = Address of the control enable CCB
151 C.NSP - Source PDV and Channel
152 C.STS - Stop modifier:
153 Bit 0 = 0 - Stop and disable
154 Bit 0 = 1 - Stop only
155 R5 = Address of the physical link access block
156
157 Registers modified:
158 R0, R1, R2, R3, R4, R5
159
160 SHUT: SAVRG <R5> ; Save address of physical link access block
161 CALL GTSLN ; Get SLN and logical tributary #
162 CALL STNMST ; Compute address of network management state byte
163 MOVB #LN.OFF,(R5) ; Set state to 'OFF'
164 RESRG <R5> ; Recover address of physical link access block
165
166 MOVB #LN.OFF,L$STT(R5)
167
168 MOV #IE.ABO,R0 ; Assume a state error
169 TST L$CTL(R5) ; Is there a pending control function?
170 BNE CCBDN ; If NE, yes ... report a state error
171 MOV R4,L$CTL(R5) ; Save address of the control enable CCB
172
173 MOV #LR$STP,R0 ; Assume only a stop is required
174 BIT #LENB,C.STS(R4) ; Is only the stop function required?
175 BNE 10$ ; If NE, yes
176 BIS #LR$DIS,R0 ; Include a disable as well
177
178 10$: CLR L$TCB(R5) ; No active user now
179 CALL FNCAB ; Abort any pending control functions
180 CALLR ABORG ; Abort current operations on the link
  
```

|                  |                  |                  |                  |                   |
|------------------|------------------|------------------|------------------|-------------------|
| LF.LPB= 001000   | L\$FNC 000014    | N\$SACC= 000001  | STRCHN= ***** GX | XV.BRO= 000001    |
| LF.MDC= 000100   | L\$LEN 000026    | N\$SBUF= 000001  | STRTC 001054P    | XV.OTH= 000000    |
| LF.MFL= 004000   | L\$LIN 000006    | N\$SLDV= 000001  | S\$SWRG= 000000  | X\$SDBT= 000000   |
| LF.MTP= 000020   | L\$IST 000000    | N\$SMCP= 000001  | S\$SYSZ= 007600  | ZF.COQ= 001000    |
| LF.PAC= 000200   | L\$LUN 000024    | N\$SMML= 000001  | S.COST 000001    | ZF.DDM= 000001    |
| LF.RDY= 040000   | L\$RCV 000020    | N\$SMOV= 000010  | S.FLG 000000     | ZF.DIA= 004000    |
| LF.REA= 010000   | L\$RSR 000002    | N\$SNCT= 000001  | S.LEN 000004     | ZF.DLC= 000002    |
| LF.SER= 000040   | L\$STT 000007    | N\$SPEM= 000001  | S.NMST 000002    | ZF.DVP= 100000    |
| LF.TIM= 000010   | L\$TCB 000022    | PLAST = ***** GX | S.OWNR 000003    | ZF.INI= 040000    |
| LF.UNL= 020000   | L\$TIM 000004    | PROSOV= 000000   | TMOUT 000332RG   | ZF.KMX= 000020    |
| LF.X2P= 000000   | L\$TIMI 000005   | P\$SP45= 000000  | T\$ASN 000016    | ZF.LLC= 000004    |
| LLCRS = ***** GX | L\$TIP 000003    | P\$SWRD= 000000  | T\$CTL 000002    | ZF.LMC= 000100    |
| LN.CLO= 000000   | L\$XMT 000016    | Q\$SOPT= 000010  | T\$DDM 000014    | ZF.MAN= 020000    |
| LN.DUM= 000005   | L\$SASG= 000000  | RCVCHR= ***** GX | T\$DLC 000015    | ZF.MFL= 000010    |
| LN.LOA= 000004   | L\$SDRV= 000000  | RCVCP 000444R    | T\$FLG 000006    | ZF.MTM= 000400    |
| LN.LOO= 000003   | L\$SP11= 000001  | RCVDN = ***** GX | T\$KRB 000020    | ZF.MUX= 000040    |
| LN.OAU= 000003   | L\$S11R= 000000  | RCVERY 000340R   | T\$LEN 000024    | ZF.PSE= 002000    |
| LN.OFF= 000001   | L.COST 000015    | RLSCHN= ***** GX | T\$LIN 000003    | ZF.SLI= 010000    |
| LN.ON = 000000   | L.CTL 000012     | RQALT = ***** GX | T\$NAM 000000    | ZF.TIM= 000200    |
| LN.OOP= 000004   | L.CVA 177776     | RQDON = ***** GX | T\$NMST 000022   | ZF.X3P= 000000    |
| LN.OPE= 000001   | L.DDM 000002     | R\$SDER= 000000  | T\$NSTA 000005   | ZS.ASN= 100000    |
| LN.REF= 000002   | L.DDS 000004     | R\$SK11= 000001  | T\$PDV 000007    | ZS.BSY= 140000    |
| LN.SER= 000002   | L.DLC 000003     | R\$SND= 000000   | T\$SFLG 000004   | Z.AVL 000014      |
| LN.STA= 000017   | L.DLM 000006     | R\$S11M= 000000  | T\$SLF 000012    | Z.DAT 000016      |
| LN.SUB= 000360   | L.DLS 000010     | SF.ACT= 000200   | T\$SLN 000010    | Z.DSP 000000      |
| LN.TRI= 000006   | L.FLG 000000     | SF.ENA= 000100   | T\$SKMG= 000000  | Z.FLG 000010      |
| LR\$CTL= 000003  | L.KRBA 000016    | SF.LPB= 000004   | T\$SMIN= 000000  | Z.LEN = 000016    |
| LR\$DEA= 000200  | L.LEN = 000022   | SF.MFL= 000040   | U.VCB = ***** GX | Z.LLN 000006      |
| LR\$DIS= 000020  | L.MPF 000022     | SF.PAC= 000020   | V\$CTR= 001000   | Z.MAP 000020      |
| LR\$STP= 000010  | L.NMST 000020    | SF.REA= 000010   | XF.ASS= 000030   | Z.NAM 000004      |
| LS\$DAT= 000004  | L.NSTA 000014    | SF.SER= 000001   | XF.DEA= 000040   | Z.PCB 000012      |
| LS\$ERR= 000200  | L.OWNR 000021    | SF.SVC= 000002   | XF.GMC= 000010   | Z.SCH 000007      |
| LS\$FRE= 000000  | L.UNT 000013     | SF.UNL= 000040   | XF.OFF= 000010   | \$BIAS = ***** GX |
| LS\$IDL= 000001  | M\$SCRB= 000124  | SHUT 000132R     | XF.ON = 000000   | \$BUFFG= ***** GX |
| LS\$MSK= 000037  | M\$SCRX= 000000  | START 000040R    | XF.REA= 000020   | \$BYTES= ***** GX |
| LS\$RST= 000020  | M\$SFCS= 000000  | STLNK = ***** GX | XF.SMC= 000000   | \$DLXCH= ***** GX |
| LS\$STP= 000010  | M\$SMGE= 000000  | STNMST= ***** GX | XF.STD= 000200   | \$DLXTB 000000RG  |
| LS\$STR= 000002  | M\$SNET= 000000  | STOPC 001162RG   | XF.STP= 000010   | \$IOS = ***** GX  |
| L\$CTL 000012    | M\$SOVR= 000000  | STPCHN= ***** GX | XMTAB = ***** GX | \$VA = ***** GX   |
| L\$ERR 000010    | NULL 000336R     | STPRQ = ***** GX | XMTCP 000352R    | .\$\$\$\$= 000034 |
| L\$FLG 000001    | NXINA = ***** GX |                  |                  |                   |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
001372 001 (RW,I,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 61  
Work file writes: 66  
Size of work file: 19564 Words ( 77 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:26.34  
SY:DLXCEX.V2,[131,134]DLXCEX/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[131,10]DLXCEX



```

175 .SBTTL Retrieve destination Ethernet address
176 +
177 **--RCVDAD-Retrieve destination Ethernet address
178
179 Retrieve the destination Ethernet address for this received message.
180
181 Inputs:
182 R0 = Address of characteristics block
183 R1 = Default status (CS.IGN = Characteristic ignored)
184 R3 = Address of the I/O packet
185 R4 = Address of the receive CCB
186 C.BUF - Bias of received message
187 C.BUF+2 - Virtual address of received message
188 C.CNT - Length of received message
189 C.NSP - Virtual address of Ethernet header
190 C.PRO - Protocol type from message
191 C.ADD - Source address from Ethernet message
192 R5 = Address of the physical link access block
193
194 Outputs:
195 R1 = Status to return for characteristic
196
197 000206 026027 000002 000006 RCVDAD: CMP C.DATI(R0),#6 ; Are there at least 6 bytes available?
198 000214 103436 BLO 10$ ; If LO, no
199
200 000216 MAP C.BUF(R4) ; Map to the message buffer
201 000224 016401 000004 MOV C.NSP(R4),R1 ; Point to the Ethernet header
202 000230 062701 000006 ADD #6,R1 ; Point past destination address
203
204 000006 .REPT 6
205 MOVB -(R1),-(SP) ; Save address on the stack
206 .ENDR
207
208 000250 MAP I.PRM+6(K3) ; Restore mapping to user's buffer
209 000256 010001 MOV R0,R1 ; Point to characteristics block
210 000260 062701 000010 ADD #C.CHRL,R1 ; Point to output buffer
211
212 000006 .REPT 6
213 MOVB (SP)+,(R1)+ ; Copy address to output buffer
214 .ENDR
215
216 000300 012760 000006 000004 MOV #6,C.DATI(R0) ; Fill in length of the address
217 000306 012701 000001 MOV #CS.SUC,R1 ; Return successful completion
218
219 000312 10$: RETURN

```

```

61      .SBTTL  LLC to LLC control functions
62      ;
63      ; Stat's values for control functions
64      ;
65      000001      $TOP    == 1      ; Partial shutdown - stop only
66      000000      OFF     == 0      ; Full shutdown - stop and disable
67      ;
68      ; Status values for start enable
69      ;
70      000004      $TRT    == 4      ; Startup the line
71      ;
72      ; For stop complete and start enable
73      ;
74      ;
75      000000      LDIS    == 0      ; Line is disabled
76      000001      LENB    == 1      ; Line is enabled
77      000002      ENBX    == 2      ; Line enable control
78      ;

```

```

474                                     .SBTTL Control complete processing
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497 000742 116405 000006
498 000746 016400 000014
499 000752 016401 000016
500 000756 016402 000020
501 000762 016403 000022
502
503 000766 105764 000012
504 000772
505
506
507
508
509 000776
510 001000
511 001002
512 001004
513 001006
514
515
516
517
518

                                     :+
                                     :*-CTLCCP-Control complete processing
                                     :
                                     :   This routine is called from the COMM/EXEC when a control complete CCB
                                     :   from another LLC has been dispatched.
                                     :
                                     :   Inputs:
                                     :       R4 = Address of the control complete CCB
                                     :           C.LIN - Index of next routine to execute
                                     :           C.TS - State of the line (enable or disable) on a stop
                                     :           C.BUF - PDV and Channel of target LLC
                                     :           C.BUF+2 - Address in target PDV channel mapping table
                                     :           C.CNT - SLN and logical tributary # of <Line-id>
                                     :           C.FLG - Address of the I/O packet
                                     :
                                     :   Outputs: (to processing routines)
                                     :       R0 = PDV and Channel of target LLC
                                     :       R1 = Address in target PDV channel mapping table
                                     :       R2 = SLN and logical tributary # of <Line-id>
                                     :       R3 = Address of the I/O packet
                                     :       The N-bit is set from a TSTB C.STS(R4)
                                     :
CTLCCP::MOVB C.LIN(R4),R5 ; Get processing routine index
          MOV C.BUF(R4),R0 ; Restore PDV and Channel of target LLC
          MOV C.BUF+2(R4),R1 ; Restore address in PDV channel mapping table
          MOV C.CNT(R4),R2 ; Restore SLN and logical tributary # of <Line-id>
          MOV C.FLG(R4),R3 ; Restore address of the I/O packet
          TSTB C.STS(R4) ; Sense the completion status
          CALLR @CCPTBL(R5) ; Dispatch to processing routine
                                     :+
                                     : Control complete dispatch table
                                     :
CCPTBL: DISP ASS,CILRE1 ; Notify target LLC of assignment
          DISP REA,CTLRE2 ; Complete the reassignment
          DISP OFF,CTLOF2 ; Complete the off
          DISP DEA,CTLDE1 ; Complete the deassignment
          DISP DON,CTLDON ; Post I/O completion on the request
                                     :
          .IF DF R$SMPL
          DISP OFL,CTLROF ; Complete reconfiguration offline request
          DISP ONL,CTLRON ; Complete reconfiguration online request
          .ENDC

```

|    |     |                                    |
|----|-----|------------------------------------|
| 5- | 44  | Macro definitions                  |
| 6- | 50  | Local data                         |
| 7- | 88  | Exec vector table                  |
| 8- | 132 | Physical link access block storage |

```
44 .SBTTL Macro definitions
45
46 .MCALL SAVRG,RESRG,MAP,$QDLC
47 .MCALL CCBDF$,PDVDF$,PLADF$,SLTDF$
48
49 000000 CCBDF$ ; Define CCB offsets
50 000000 PDVDF$ ; Define PDV offsets
51 000000 PLADF$ ; Define PLA offsets
52 000000 SLTDF$ ; Define SLT offsets
```

1 .IIF NDF M\$\$ACP .TITLE DLXQ10 - ISSUE I/O TO DLX  
2 .IIF DF M\$\$ACP .TITLE DLXAST - ISSUE I/O TO DLX AND WAIT FOR AST  
3 .IDENT /V05.00/  
4 .NLIST BEX

5  
6 : COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
7 : DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

8  
9 : THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
10 : ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
11 : INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
12 : COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
13 : OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
14 : TRANSFERRED.

15 : THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
16 : AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
17 : CORPORATION.

18 : DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
19 : SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

20 :  
21 : MODULE DESCRIPTION:  
22

23 : NETWORK MANAGEMENT - ISSUE I/O REQUESTS TO NX: (DLX)  
24

25 : DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING  
26

27 : IDENT HISTORY:  
28

29 : 1.00 24-MAR-81  
30 : CREATED FROM LICHOP, CONDITIONALIZED TO FORK WHEN RUNNING IN  
31 : THE NETWORK MANAGEMENT ACP  
32

33 : 2.C 16-APR-82  
34 : DECNET-11M V3.1  
35 : DECNET-11M-PLUS V1.1  
36

37 : 4.00 07-NOV-83  
38 : DECNET-11M V4.0  
39 : DECNET-11M-PLUS V2.0  
40

41 : 5.00 22-JUL-85  
42 : DECnet-11M/S V4.2  
43 : DECnet-11M-Plus V3.0  
44 : DECnet-Micro/RSX V1.0  
45  
46  
47  
48  
49  
50  
51  
52  
53

DLXQIO      CREATED BY    MACRO    ON 29-JUN-85 AT 12:23      PAGE 2      M 11  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE    | REFERENCES                          |
|---------|----------|-------------------------------------|
| LS\$MRT | = 000020 | #6-75                               |
| LS\$MWN | = 000040 | #6-75                               |
| LS\$MXB | = 001000 | #6-75                               |
| LS\$MXR | = 000020 | #6-75      6-75      6-75      6-75 |
| LS\$MXW | = 000040 | #6-75      6-75      6-75      6-75 |
| LS\$NMT | = 020000 | #6-75                               |
| LS\$NOR | = 100000 | #6-75                               |
| LS\$NTI | = 000200 | #6-75                               |
| LS\$NTL | = 000001 | #6-75                               |
| LS\$NUM | = 000100 | #6-75      6-75                     |
| LS\$OWN | = 000010 | #6-75                               |
| LS\$PLT | = 004000 | #6-75                               |
| LS\$PRO | = 010000 | #6-75                               |
| LS\$PVC | = 020053 | #6-75                               |
| LS\$RET | = 000200 | #6-75      6-75      6-75           |
| LS\$SER | = 100000 | #6-75                               |
| LS\$STA | = 000002 | #6-75                               |
| LS\$SVC | = 000362 | #6-75                               |
| LS\$TAD | = 000020 | #6-75                               |
| LS\$TH1 | = 000100 | #6-75                               |
| LS\$TH2 | = 000200 | #6-75                               |
| LS\$TH3 | = 000400 | #6-75                               |
| LS\$XMT | = 040000 | #6-75                               |
| L\$ACT  | 000104   | #6-75                               |
| L\$BBT  | 000114   | #6-75                               |
| L\$BLK  | 000262   | #6-75                               |
| L\$BSA  | 000122   | #6-75                               |
| L\$BSD  | 000123   | #6-75                               |
| L\$BSI  | 000124   | #6-75                               |
| L\$BUF  | 000060   | #6-75                               |
| L\$CHN  | 000054   | #6-75                               |
| L\$CMB  | 000234   | #6-75                               |
| L\$COS  | 000101   | #6-75                               |
| L\$CTB  | 000053   | #6-75                               |
| L\$CTL  | 000002   | #6-75                               |
| L\$CUS  | 000260   | #6-75                               |
| L\$DDM  | 000000   | #6-75                               |
| L\$DDT  | 000106   | #6-75                               |
| L\$DEA  | 000105   | #6-75                               |
| L\$DLT  | 000110   | #6-75                               |
| L\$DTE  | 000240   | #6-75                               |
| L\$DTEL | 000236   | #6-75                               |
| L\$DTEP | 000250   | #6-75                               |
| L\$FLG  | 000064   | #6-75                               |
| L\$FLX  | 000076   | #6-75                               |
| L\$FL1  | 000066   | #6-75                               |
| L\$FL2  | 000070   | #6-75                               |
| L\$HBT  | 000254   | #6-75                               |
| L\$HTM  | 000226   | #6-75                               |
| L\$INA  | 000125   | #6-75                               |
| L\$IND  | 000126   | #6-75                               |
| L\$INI  | 000127   | #6-75                               |

```
271 .SBTTL Validate characteristics buffer
272
273 *--CHKCHR-Validate characteristics buffer
274
275 Validate that the characteristics buffer is in the correct format.
276
277 Inputs:
278 R0 = Offset in I/O packet to characteristics buffer
279 R3 = Address of the I/O packet
280
281 Outputs:
282 'C' Clear - Characteristics buffer format is valid
283 'C' Set - Characteristics buffer format is invalid
284
285 000434 CHKCHR::SAVRG <R1> ; Get a free register
286 000436 060300 ADD R3,R0 ; Compute address of buffer descriptor
287 000440 MAP (R0)+ ; Map to the characteristics buffer
288 000444 012001 MOV (R0)+,R1 ; Get virtual address of the buffer
289 000446 011000 MOV (R0),R0 ; Get # of bytes in the buffer
290
291 000450 005700 10$: TST R0 ; Any more bytes left in the buffer?
292 000452 001423 BEQ 30$ ; If EQ, no ... buffer is valid
293 000454 020027 000010 CMP R0,#C.CHRL ; Enough space for a characteristics descriptor?
294 000460 103417 BLO 20$ ; If LO, no
295 000462 032761 000001 000002 BIT #1,C.DATI(R1) ; Next characteristics block word aligned?
296 000470 001013 BNE 20$ ; If NE, no
297
298 000472 005061 000006 CLR C.STAT(R1) ; Clear characteristics
299 000476 166100 000002 SUB C.DATI(R1),R0 ; Remove space for this characteristic
300 000502 162700 000010 SUB #C.CHRL,R0 ; and the common overhead
301 000506 066101 000002 ADD C.DATI(R1),R1 ; Point to next characteristic descriptor
302 000512 062701 000010 ADD #C.CHRL,R1 ;
303 000516 000754 BR 10$ ; and loop
304
305 000520 000261 20$: SEC ; Indicate error
306 000522 30$: RESRG <R1>
307 000524 RETURN
```



```
794 .SBTTL Find <Line-id> in the COMM/EXEC database
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816 002026
817
818
819
820
821
822 002034 005005
823
824 002036 005067 000000G
825 002042 005067 000002G
826 002046 005067 000004G
827 002052 005067 000010G
828
829 002056
830
831 002064 016300 000002G
832 002070 016304 000004G
833 002074 060004
834
835 002076 012703 000055
836 002102
837 002106 103456
838 002110 010167 000000G
839 002114 020004
840 002116 103050
841
842 002120 010102
843 002122
844 002126 103407
845 002130 067702 000000G
846 002134 011202
847 002136 132762 000040 000010
848 002144 001003
849
850 002146 012703 000056 10$: MOV #',R3 ; Set new terminating character

      .SBTTL Find <Line-id> in the COMM/EXEC database
      **--FNDLN-Find <Line-id> in the COMM/EXEC database
      Parse the user supplied <Line-id> string and scan the COMM/EXEC
      databases to find the <Line-id>.
      -
      Inputs:
      R3 = Address of I/O packet
      I.PRM - Bias of <Line-id> string
      I.PRM+2 - Virtual address of <Line-id> string
      I.PRM+4 - Length of <Line-id> string
      Outputs:
      R1 = DDM: Address of system line table
      DLC or LLC: Address of process PDV
      'C' Clear - <Line-id> found in databases
      'C' Set - <Line-id> not present in databases
      Registers modified:
      R0, R2
      FNDLN:: SAVRG <R3,R4,R5> ; Save some registers
      .IF DF R$$MPL
      SAVMAP ; Save current mapping
      .ENDC
      CLR R5 ; Initialise multiplexor flag
      CLR $PBLK+T$NAM ; Initialise parameter block
      CLR $PBLK+T$CTL ; ...
      CLR $PBLK+T$SFLG ; ...
      CLR $PBLK+T$SLN ; ...
      MAP I.PRM(R3) ; Map to the <Line-id> string
      MOV I.PRM+2(R3),R0 ; Get virtual address of the string
      MOV I.PRM+4(R3),R4 ; Compute address of the end of string +1
      ADD R0,R4 ; ...
      MOV #'-,R3 ; Set up allowed terminating character
      CALL CAT5 ; Convert process name to RAD50
      BCS 50$ ; If CS, illegal character in process name
      MOV R1,$PBLK+T$NAM ; Save process name
      JMP R0,R4 ; At end of the string?
      BHIS 40$ ; If HIS, yes ... look for <Line-id>
      MOV R1,R2 ; Find PDV index for this process
      CALL @PDVID ; ...
      BCS 10$ ; If CS, assume non-multiplexor
      ADD @PDVTA,R2 ; Get pointer to PDV
      MOV (R2),R2 ; ...
      BITB #ZF,MUX,Z.FLG(R2)
      BNE 20$ ; If NE, device is a multiplexor
      MOV #',R3 ; Set new terminating character
```

DLXSUB      CREATED BY    MACRO    ON 28-JUN-85 AT 22:56      PAGE 2      M 14  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|---|
| LF.SER  | = 000040   | #5-50   |
| LF.TIM  | = 000010   | #5-50   |
| LF.UNL  | = 020000   | #5-50      16-607   |
| LF.X2P  | = 000000   | #5-50   |
| LLCTA   | = ***** GX | 11-331      15-513      18-782  |
| LN.CLO  | = 000000   | #5-50   |
| LN.DUM  | = 000005   | #5-50   |
| LN.LOA  | = 000004   | #5-50   |
| LN.LOO  | = 000003   | #5-50   |
| LN.OAU  | = 000003   | #5-50   |
| LN.OFF  | = 000001   | #5-50   |
| LN.ON   | = 000000   | #5-50      6-109  |
| LN.OOP  | = 000004   | #5-50   |
| LN.OPE  | = 000001   | #5-50   |
| LN.REF  | = 000002   | #5-50   |
| LN.SER  | = 000002   | #5-50   |
| LN.STA  | = 000017   | #5-50      6-108  |
| LN.SUB  | = 000360   | #5-50   |
| LN.TRI  | = 000006   | #5-50   |
| L\$LEN  | 000026     | 25-1034   |
| L\$LIN  | 000006     | 21-935      *25-1038  |
| L.COST  | 000015     | #5-50   |
| L.CTL   | 000012     | #5-50      18-740   |
| L.CVA   | 177776     | #5-50   |
| L.DDM   | 000002     | #5-50      18-738      18-750   |
| L.DDS   | 000004     | #5-50   |
| L.DLC   | 000003     | #5-50   |
| L.DLM   | 000006     | #5-50   |
| L.DLS   | 000010     | #5-50   |
| L.FLG   | 000000     | #5-50      26-1058      26-1060   |
| L.KRBA  | 000016     | #5-50      18-751   |
| L.LEN   | = 000022   | #5-50   |
| L.MPF   | 000022     | #5-50      *6-102      *12-387      12-390      15-530      15-531      18-769      18-770      26-1070 |
| L.NMST  | 000020     | #5-50      18-752      26-1056      26-1058      26-1060      26-1062      26-1070                      |
| L.NSTA  | 000014     | #5-50      6-94      12-378      12-389      15-508      15-524      18-758      18-763      26-1062    |
| L.DWNR  | 000021     | #5-50   |
| L.UNT   | 000013     | #5-50      18-742   |
| MAPAD   | 002354     | RG      #22-956      23-989   |
| MAXOV   | = ***** GX | 14-451      14-452  |
| N\$SVCT | = *****    | 10-287      19-829  |
| PDVID   | = ***** GX | 18-719      19-843      20-904  |
| PDVTA   | = ***** GX | 17-673      18-723      19-845      20-908      21-932      22-962                                      |
| PLAST   | = ***** GX | 25-1037   |
| PRCLD   | 002416     | RG      #23-988   |
| RDBRT   | = ***** GX | 24-1006   |
| RELOC   | = ***** GX | 7-157   |
| RLSCHN  | 002444     | RG      14-486      #24-1005      24-1008      19-878   |
| R\$MPL  | = *****    | 16-610      16-625      19-818  |
| R\$1TD  | = *****    | 5-48  |
| R\$1TM  | = 000000   | 5-48  |
| R\$1TS  | = *****    | 5-48  |
| SF.ACT  | = 000200   | #5-50      6-102      12-387      12-391  |

M 15

EVL - EVENT LOGGER PROCESS      MACRO V05.03b Friday 28-Jun-85 22:58    Page 9-1

LOG NETWORK EVENT

```

173 000156 016453 000006 000024      10$:  MOV     C.LIN(R4),E.LIN(R3)
174 000164      MAP     C.BUF(R4)          ; MAP TO THE DATA AREA
175 000172 016404 000016      MOV     C.BUF+2(R4),R4      ; SET UP BUFFER POINTER
176
177 000176 132763 000002 000020      20$:  BITB    #EV.MAP,E.CTL(R3)
178 000204 001402      BEQ     30$          ; IF EQ, LINE-ID ALREADY MAPPED
179 000206      CALL     MAPCHN          ; MAP PDV & CHANNEL TO SLN & STATION
180
181 000212 132763 000200 000020      30$:  BITB    #EV.PRT,F.CIL(R3)
182 000220 001402      BEQ     35$          ; IF EQ, NO PORT ASSOCIATED WITH THIS EVENT
183 000222      CALL     MAPORT          ; MAP PORT TO PVC/DLM CIRCUIT
184
185 000226 116102 000001      35$:  MOVB    1(R1),R2          ; ISOLATE BYTE COUNT
186 000232 042702 177400      BIC     #^C<377>,R2          ;
187 000236 020227 000150      CMP     R2,#E.LEN-E.DATA      ; IS THERE TOO MUCH DATA?
188 000242 101402      BLOS    40$          ; IF LOS, NO
189 000244 012702 000150      MOV     #E.LEN-E.DATA,R2      ; USE OUR MAXIMUM
190 000250 010263 000022      40$:  MOV     R2,E.SIZ(R3)          ; FILL IN SIZE OF ADDITIONAL DATA
191 000254 001406      BEQ     60$          ; IF EQ, NONE PRESENT
192
193 000256 010300      MOV     R3,R0          ; POINT TO DATA AREA
194 000260 062700 000046      ADD     #E.DATA,R0          ; ...
195
196 000264 112420      50$:  MOVB    (R4)+(R0)+      ; COPY DATA TO BUFFER
197 000266      SOB     R2,50$          ; ...
198 000272      60$:
199 000272      INHIB$          ; INHIBIT INTERRUPTS
200
201      .IF DF R$$MPL
202      .IF NDF R$$PRO
203
204      BIT     #F2.MP,@FMSK2      ;::IS THIS A MULTI-PROCESSOR
205      BEQ     62$          ;::BR IF NO
206      CALL    @MPLCK          ;::INTERLOCK COMM/EXEC RESOURCES
207
208      62$:
209      .ENDC
210
211 000304      CALL    QUEUE          ;::PLACE EVENT ON COLLECTION QUEUE
212
213      .IF DF R$$MPL
214      .IF NDF R$$PRO
215
216      BIT     #F2.MP,@FMSK2      ;::IS THIS A MULTI-PROCESSOR
217      BEQ     64$          ;::BR IF NO
218      CALL    @ (SP)+          ;::CO-ROUTINE RETURN TO UNLOCK RESOURCES
219
220      64$:
221      .ENDC
222      .ENDC
223
224 000310      ENABL$          ;::ENABLE INTERRUPT AGAIN
225 000314 016500 000004      MOV     E$TCB(R5),R0      ; UNSTOP COLLECTOR TASK
226
227      .IF DF R$$MPL
228      .IF NDF R$$PRO
229
230      BIT     #F2.MP,@FMSK2      ;::IS THIS A MULTI-PROCESSOR?

```



```

443 001004 010200          20$:  MOV    R2,R0          ; Compute address of alias name
444 001006 062700 000002      ADD    #A.NAM,R0          ;
445 001012          CALL    CHKNAM          ; Check for name match
446 001016 001360          BNE    10$          ; If NE, no match
447
448 001020 062702 000012      ADD    #A.REM,R2          ; Point to remote node name
449 001024 012700 000000G     MOV    #SDSNOD,R0          ; Point to internal node name
450
451          000033          .REPT    3
452          MOV    (R2)+,(R0)+          ; Copy remote node name
453          .ENDR
454
455 001036 016700 000000G     MOV    $REQID,R0          ; Did user specify any of
456 001042 056700 000000G     BIS    $PASSW,R0          ; requestor id, password
457 001046 056700 000000G     BIS    $ACCN1,R0          ; or accounting info
458 001052 001030          BNE    80$          ; Then leave those fields alone
459
460 001054 012700 000000G     MOV    #REQID,R0          ;
461 001060 112203          MOV    (R2)+,R3          ; Is requestor id present in alias block?
462 001062 001404          BEQ    40$          ; If EQ, no
463 001064 010320          MOV    R3,(R0)+          ; Fill in requestor id length
464 001066 112220          30$:  MOVB   (R2)+,(R0)+          ; Move requestor id into connect block
465 001070          SOB    R3,30$          ;
466
467 001074 012700 000000G     MOV    #PASSW,R0          ;
468 001100 112203          MOVB   (R2)+,R3          ; Is password present in alias block
469 001102 001404          BEQ    60$          ; If EQ, no
470 001104 010320          MOV    R3,(R0)+          ; Fill in password length
471 001106 112220          50$:  MOVB   (R2)+,(R0)+          ; Move password into connect block
472 001110          SOB    R3,50$          ;
473
474 001114 012700 000000G     MOV    #ACCN1,R0          ;
475 001120 112203          MOVB   (R2)+,R3          ; Is account info present in alias block
476 001122 001404          BEQ    80$          ; If EQ, no
477 001124 010320          MOV    R3,(R0)+          ; Fill in account info length
478 001126 112220          70$:  MOVB   (R2)+,(R0)+          ; Move account info into connect block
479 001130          SOB    R3,70$          ;
480
481 001134 126727 000000G 000040 80$:  CMPB   $DSNOD,#'          ; Blank node name
482 001142 001442          BEQ    100$          ; Means connect to local system
483
484 001144 010402          MOV    R4,R2          ; Point to the remote node listhead
485 001146 062702 000002      ADD    #DSRNN,R2          ; Point to remote name listhead
486 001152 011246          90$:  MOV    (R2),-(SP)          ; Get next entry in list
487 001154          CALL    @CEACC          ; Gain access to it
488 001160 012602          MOV    (SP)+,R2          ; Get virtual address of remote name block
489 001162 001433          BEQ    110$          ; If EQ, no more remote name blocks
490
491 001164 010200          MOV    R2,R0          ; Compute address of node name
492 001166 062700 000002      ADD    #R.NAM,R0          ;
493 001172          CALL    CHKNAM          ; Check for name match
494 001176 001365          BNE    90$          ; If NE, no match
495 001200 032762 100000 000012  BIT    #RF.LOD,R.FLAG(R2) ; Is it a loopback node name ?
496 001206 001403          BEQ    95$          ; If EQ, no
497 001210 152767 000001 000000G  BITB   #1,$FLAGS          ; Else, yes - indicate to caller
498 001216 016265 000010 000012 95$:  MOV    R,ALD(R2),N$SNOD(R5)
499 001224 032765 176000 000012 97$:  BIT    #NM$ACC,N$SNOD(R5)

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES         |
|---------|------------|--------------------|
| R\$SEIS | = *****    | 20-675             |
| R.ADD   | 000010     | #6-56 14-498       |
| R.FLAG  | 000012     | #6-56 14-495       |
| R.LEN   | 000014     | #6-56              |
| R.LNK   | 000000     | #6-56              |
| R.NAM   | 000002     | #6-56 14-492       |
| SAVOPT  | = ***** GX | 8-132 9-205 16-550 |
| SETREQ  | 000000 R   | #7-78 8-130 9-207  |
| SNDACC  | 000056 RG  | #8-129             |
| SNDCON  | 000170 RG  | #9-173             |
| SNDDIS  | 001300 PG  | #16-547            |
| SNSESD  | = ***** GX | 8-145 9-223        |
| ST\$CC  | = 000004   | 8-146              |
| ST\$CIS | = 000002   | 9-202              |
| ST\$DIP | = 000012   | 16-547 16-551      |
| TRMLNK  | = ***** GX | 16-553             |
| T\$FLAG | 000044     | #6-49              |
| T\$LIF  | 000013     | #6-49              |
| T\$LIFL | 000013     | #6-49              |
| T\$LIFO | 000013     | #6-49              |
| T\$LIFS | 000013     | #6-49              |
| T\$LIN  | 000000     | #6-49              |
| T\$LIPS | 000006     | #6-49              |
| T\$LLD  | 000012     | #6-49              |
| T\$LLDC | 000045     | #6-49              |
| T\$LLDL | 000012     | #6-49              |
| T\$LLDO | 000012     | #6-49              |
| T\$LLDS | 000012     | #6-49              |
| T\$LLEN | 000046     | #6-49              |
| T\$LOPR | 000002     | #6-49              |
| T\$LTCL | 000024     | #6-49              |
| T\$LTIM | 000026     | #6-49              |
| T\$LTPR | 000014     | #6-49              |
| T\$LTPS | 000020     | #6-49              |
| T\$NAPL | 000004     | #6-49              |
| T\$NFE  | 000000     | #6-49              |
| T\$NLEN | 000010     | #6-49              |
| T\$NNUL | 000002     | #6-49              |
| T\$NOPL | 000006     | #6-49              |
| T\$NRNI | 000042     | #5-49              |
| T\$NRPL | 000005     | #6-49              |
| T\$NRUL | 000007     | #6-49              |
| T\$NVR  | 000001     | #6-49              |
| T\$RPRI | 000040     | #6-49              |
| T\$SVC  | 000034     | #6-49              |
| T\$T5   | 000030     | #6-49              |
| T\$T6   | 000032     | #6-49              |
| T.UCB   | = ***** GX | 14-430             |
| UISAR6  | = ***** GX | 14-406             |
| USRINT  | 001346 RG  | #17-573            |
| US\$DSC | = 000004   | 16-552             |
| W.LUN   | 000003     | 17-578             |

```

182                                     .SBTTL Timeout processing
183
184 :+ *-TMOUT-Timeout processing
185
186 : This routine is entered once per second from the COMM/EXEC.
187
188 : Registers modified:
189 : R0, R1, R2, R3, R4, R5
190
191 000232 012705 000000G TMOUT:: MOV #PLAST,R5 ; Set up address of first physical link access block
192 000236 116746 000000G MOVB $DLXCH,-(SP) ; Set up count of channels
193
194 000242 011501 10$ MOV (R5),R1 ; Get link state and flags
195 000244 100014 BPL 40$ ; If PL, no resource recovery needed
196 000246 042715 100000 BIC #L$RSR,(R5) ; Clear resource recovery fla
197 000252 042701 177740 BIC #C<L$MSK>,R1 ; Isolate link state
198 000256 001407 BEQ 40$ ; If EQ, physical link block is not in use!
199
200 000260 012702 000340' MOV #RCVERY,R2 ; Point to recovery dispatch table
201 000264 006201 20$ ASR R1 ; Find first bit set
202 000266 103402 BCS 30$ ; If CS, dispatch
203 000270 005722 TST (R2)+ ; Skip over this entry
204 000272 000774 BR 20$ ; and try again
205 000274 CALL @ (R2)+ ; Dispatch to recover routine
206
207 000276 105765 000004 40$ TSTB L$TIM(R5) ; Is the receive timer running?
208 000302 001410 BEQ 50$ ; If EQ, no
209 000304 105365 000004 DECB L$TIM(R5) ; Count down the receive timer
210 000310 001005 BNE 50$ ; If NE, still active
211
212 000312 012700 000000C MOV #IE.TMO&377,R0 ; Complete a receive with a timeout
213 000316 005001 R1 ;
214 000320 CLR R1 ;
215 CALL RCVDN ;
216
217 000324 062705 000026 50$ ADD #L$LEN,R5 ; Point to next physical link access block
218 000330 105316 DECB (SP) ; Any more to go?
219 000332 001343 BNE 10$ ; If NE, yes
220
221 000334 005726 TST (SP)+ ; Clean up the stack
222 NULL: RETURN
223
224 :+
225 : Resource recovery dispatch table
226
227 000340 000000G RCVERY: .WORD STRCHN ; IDLE - Try to start channel
228 000342 000336' .WORD NULL ; STARTING
229 000344 000000G .WORD ITRANS ; DATA
230 000346 000000G .WORD STPCHN ; STOPPING - Try to stop channel
231 000350 000000G .WORD STPCHN ; RESTARTING - Try to stop channel

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES                              |
|---------|-------------|---|
| ABORQ   | = ***** GX  | 10-180                                  |
| ASYNCH  | = 001012 R  | 14-374 #15-399                          |
| BLXIO   | = ***** GX  | 13-311                                  |
| CB.RDB  | = 000004    | 12-258                                  |
| CCBDN   | = 000106 R  | #9-137 10-170 19-542                    |
| CCBRT   | = ***** GX  | 12-264 14-366                           |
| CE.DIS  | = 100366    | 15-599                                  |
| CHARC   | = 001140 R  | 14-379 14-380 #18-473                   |
| CRC\$OV | = 000000    | #5-51                                   |
| CTLCCP  | = ***** GX  | 14-369                                  |
| CTLCP   | = 000706 RG | 7-75 #14-354                            |
| CTLDN   | = ***** GX  | 19-534                                  |
| CTLDSP  | = 000032 R  | 8-89 #8-91                              |
| CTLEN   | = 000022 RG | 7-70 #8-88                              |
| CTLTL   | = 000762 R  | 14-367 #14-374                          |
| C.BID   | = 000003    | 12-258                                  |
| C.BUF   | = 000014    | 13-303 13-304                           |
| C.BUF1  | = 000014    | 14-363                                  |
| C.BUF2  | = 000024    | 14-364                                  |
| C.CNT   | = 000020    | 12-251 13-294                           |
| C.FNC   | = 000010    | *9-138                                  |
| C.LIN   | = 000006    | *9-137 14-354                           |
| C.MOD   | = 000011    | 14-358                                  |
| C.NSP   | = 000004    | 9-137 12-252                            |
| C.STS   | = 000012    | 9-112 9-116 *9-139 10-174 12-250 14-362 |
| DISRQ   | = ***** GX  | 15-405                                  |
| DQPKT   | = ***** GX  | 17-448 18-475                           |
| DSLIN   | = ***** GX  | 19-531                                  |
| ENBLC   | = 001040 R  | 14-384 #16-421                          |
| ENBX    | = ***** GX  | 9-112                                   |
| FC.CCP  | = 000020    | 9-138                                   |
| FNCAB   | = ***** GX  | 10-179                                  |
| GTSIN   | = ***** GX  | 9-120 10-161                            |
| IE.ABO  | = ***** GX  | 10-168                                  |
| IE.DAD  | = ***** GX  | 13-298 13-331                           |
| IE.TMO  | = ***** GX  | 11-212                                  |
| IE.VER  | = ***** GX  | 12-257                                  |
| IO.XCH  | = 017400    | #5-51 5-51 5-51                         |
| IO.XCL  | = 016210    | #5-51                                   |
| IO.XGC  | = 017410    | #5-51                                   |
| IO.XHG  | = 016000    | #5-51                                   |
| IO.XIN  | = 016400    | #5-51                                   |
| IO.XOP  | = 015400    | #5-51                                   |
| IO.XRC  | = 015000    | #5-51                                   |
| IO.XSC  | = 017400    | #5-51                                   |
| IO.XSP  | = 017010    | #5-51                                   |
| IO.XST  | = 017000    | #5-51                                   |
| IO.XTL  | = 017000    | #5-51 5-51 5-51                         |
| IO.XTM  | = 014400    | #5-51                                   |
| IO.XTR  | = 016000    | #5-51 5-51 5-51                         |
| IS.SUC  | = *****     | 9-124 12-254 13-287 17-450              |
| ITRANS  | = ***** GX  | 11-228 17-457                           |



```

221 .SBTTL Process transmit auxilliary characteristics
222
223 +*-XMTCHR-Process transmit auxilliary characteristics
224
225 Process the characteristics present in the auxilliary characteristics
226 buffer on the transmit request.
227
228 Inputs:
229 R3 = Address of the I/O packet
230 I.PRM+6 - Bias of the characteristics buffer
231 I.PRM+10- Virtual address of the characteristics buffer
232 I.PRM+12- Length of the characteristics buffer
233 I.PRM+14- Virtual address of byte past end of characteristics buffer
234 The characteristics buffer is mapped.
235 R5 = Address of the physical link access block
236
237 Registers modified:
238 R0, R1, R2
239
240 000314 016300 000010G XMTCHR:MOV I.PRM+10(R3),R0 ; Point to start of characteristics buffer
241
242 000320 012701 000002 10$: MOV #CS.IGN,R1 ; Assume characteristic will be ignored
243 000324 011002 (R0),R2 ; Get next characteristics type code
244 000326 162702 000100 SUB #CC.ADR,R2 ; Normalise for DLX functions
245 000332 002406 BLT 20$ ; If LT, not one of our characteristics
246 000334 006302 ASL R2 ; Form word offset
247 000336 020227 000004 CMP R2,#XMMAX ; In range?
248 000342 103002 BHS 20$ ; If HIS, no
249 000344 CALL @XMTTBL(R2) ; Dispatch to processing routine
250
251 000350 010160 000006 20$: MOV R1,C.STAT(R0) ; Return status of this characteristic
252 000354 066000 000002 ADD C.DAT1(R0),R0 ; Point to next characteristic
253 000360 062700 000010 ADD #C.CHRL,R0 ; ...
254 000364 020063 000014G CMP R0,I.PRM+14(R3) ; Are we done yet?
255 000370 103753 BLO 10$ ; If LO, no
256 000372 RETURN
257
258 +*
259 Transmit characteristics dispatch table
260
261 000374 000400' XMTTBL: .WORD XMTADR ; Define Ethernet address for transmit
262 000376 000446' .WORD XMTPRO ; Define Ethernet protocol for transmit
263 000004 XMMAX =.-XMTTBL

```

```

80
81
82
83
84
85 000000 000000G      .SBTTL Dispatch tables
86 000002 000000G      ;+
87 000004 000000G      ; Dispatch table to search for <Line-id>
88 000006 000000G      ; -
89 000010 000000G      SRCTBL: .WORD FINDL      ; Turn <Line-id> on
90 000012 000000G      .WORD FINDL      ; Turn <Line-id> off
91 000014 000000G      .WORD FINDC      ; Reassign <Line-id> to another LLC
92 000016 000000G      .WORD FINDC      ; Assign <Line-id> to an LLC
93 000020 000000G      .WORD FINDC      ; Deassign <Line-id> from an LLC
94
95          000022      .WORD FINDC      ; Set owner for <Line-id>
96
97
98
99
100
101 000022 000130'      SFMAX =.-SRCTBL
102 000024 000500'      ;+
103 000026 000232'      ; Dispatch table for function processing
104 000030 000404'      ; -
105 000032 000532'      FNCTBL: .WORD CTLOK      ; Turn <Line-id> on
106 000034 000604'      .WORD CTLOFF     ; Turn <Line-id> off
107 000036 000646'      .WORD CTLREA     ; Reassign <Line-id> to another LLC
108 000040 000654'      .WORD CTLASS     ; Assign <Line-id> to an LLC
109 000042 000172'      .WORD CTLDEA     ; Deassign <Line-id> from an LLC
      .WORD CTLOWN     ; Set owner for <Line-id>
      .WORD CTLIST     ; Set line state for <Line-id>
      .WORD CTLSST     ; Set line substate for <Line-id>
      .WORD CTLSER     ; Reassign <Line-id> for service access

```

```

520 .SBTTL Continue with reassignment
521
522 ;+
523 **--CTLRE1-Continue with reassignment
524
525 When the current owning LLC has completed the STOP request, perform the
526 line reassignment and inform the target LLC.
527
528 -Inputs:
529 R0 = PDV and Channel of target LLC
530 R1 = Address in target PDV channel mapping table
531 R2 = SLN and logical tributary # of <Line-id>
532 R3 = Address of the I/O packet
533 The N-bit is set from a TSTB C.STS(R4)
534
535 CTLRE1: BMI 10$ ; If MI, an error has occurred
536
537 001010 100433 BIS #100000,R2 ; Set flag to leave network management state alone
538 001012 052702 100000 CALL ASSIGN ; Modify system databases
539 001016 016400 000024 MOV C.BUF2(R4),R0 ; Get previous PDV and Channel assignment
540 001022 012711 100000 CALL MAPAD ; Compute address in channel mapping table
541 001026 012711 100000 MOV #ZS.ASN,(R1) ; Mark the channel free
542
543 001036 112764 000002 000004 MOVB #NX.REA,C.NSP(R4)
544 001044 016464 000014 000006 MOV C.BUF(R4),C.LIN(R4)
545 001052 012764 001006 000010 MOV #FC.CTL!FS.STR,C.FNC(R4)
546 001060 042764 177776 000012 BIC #^C.LENB>,C.STS(R4)
547 001066 052764 000002 000012 BIS #ENBX,C.STS(R4) ; Format CCB to inform new owning LLC
548 001074 000002 000012 CALLR QCCBL ; Queue CCB to LLC
549
550 001100 042711 040000 10$: BIC #40000,(R1) ; Remove reservation flag from channel mapping table
551 001104 000416 BR CTLAB0 ; and abort the request

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

```
.TITLE DLXDAT - DLX data bases
.IDENT /V05.00/
.ENABL LC

: Copyright (C) 1982, 1983, 1985 by
: Digital Equipment Corporation, Maynard, MASS.
:
: This software is furnished under a license for use only on a
: single computer system and may be copied only with the
: inclusion of the above copyright notice. This software, or
: any other copies thereof, may not be provided or otherwise
: made available to any other person except for use on such
: system and to one who agrees to these license terms. Title
: to and ownership of the software shall at all times remain
: in DEC.
:
: The information in this document is subject to change without
: notice and should not be construed as a commitment by Digital
: Equipment Corporation.
:
: DEC assumes no responsibility for the use or reliability of
: its software on equipment which is not supplied by DEC.
:
: Module description
:
:     DLX data bases
:
: Distributed Systems Software Engineering
:
: Ident history:
:
: 4.00 07-NOV-83
:       DECNET-11M V4.0
:       DECNET-11M-PLUS V2.0
:
: 5.00 22-JUL-85
:       DECnet-11M/S V4.2
:       DECnet-11M-Plus V3.0
:       DECnet-Micro/R SX V1.0
:
:
```

```

54                                     .SBTTL Issue control request to DLC
55                                     :+
56                                     : **--DOFNC-Issue control function to DLC
57                                     :
58                                     : Issue a control function down to the DLC associated with a physical
59                                     : link and set up for resource recovery if necessary.
60                                     : -
61                                     : Inputs:
62                                     : R0 = Control function to be issued
63                                     : R5 = Address of the physical link access block
64                                     :
65                                     : Outputs:
66                                     : 'C' Clear - Request successfully issued
67                                     : 'C' Set - Resource allocation (CCB) failure
68                                     :
69 000000 DOFNC:: CALL @CCBG7 ; Allocate a CCB
70 000004 103415 BCS 10$ ; If CS, allocation failure
71
72 000006 010064 000010 MOV R0,C.FNC(R4) ; Set up requested function code
73 000012 016564 000006 MOV L$LIN(R5),C.LIN(R4)
74 000020 012764 000001 000006 MOV #CF.CHN,C.FLG(R4)
75 000026 105265 000002 000022 INCB L$RSR(R5) ; Count pending control request
76 000032 $QDLC ; Queue request to DLC
77 000036 RETURN
78
79 000040 052715 100000 10$: BIS #LF$RSR,(R5) ; Mark resource recovery required
80 000044 RETURN
  
```

DLXQIO - ISSUE I/O TO DLX  
MACRO CALLS AND LOCAL DATA

MACRO V05.03b Saturday 29-Jun-85 12:25 Page 5

N.10

55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68

000000

```
.SBTTL MACRO CALLS AND LOCAL DATA
;
; MACRO LIBRARY CALLS
;
; .MCALL SAVRG,RESRG,CICCX$
; .MCALL ALUN$$,WSIG$$,QIOW$$,QIO$$,GLUN$$
;
; LOCAL DATA
;
; .IF NDF M$$ACP
IOSB: .BLKW 2 ; I/O STATUS BLOCK FOR DLX QIO
;
; .ENDC ;NDF M$$ACP
```

DLXQIO      CREATED BY MACRO ON 29-JUN-85 AT 12:23      PAGE 3      N 11  
 SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE    | REFERENCES |
|---------|----------|------------|
| L\$LC   | 000224   | #6-75      |
| L\$LEN  | 000264   | #6-75      |
| L\$LM   | 000234   | #6-75      |
| L\$LM   | 000230   | #6-75      |
| L\$MRT  | 000253   | #6-75      |
| L\$MSG  | 000056   | #6-75      |
| L\$MTYP | 000075   | #6-75      |
| L\$MWN  | 000252   | #6-75      |
| L\$MXB  | 000133   | #6-75      |
| L\$MXR  | 000253   | #6-75      |
| L\$MXW  | 000252   | #6-75      |
| L\$NAM  | 000000   | #6-75      |
| L\$NLEN | = 000030 | #6-75      |
| L\$NMT  | 000116   | #6-75      |
| L\$NTI  | 000256   | #6-75      |
| L\$NTL  | 000134   | #6-75      |
| L\$NUM  | 000240   | #6-75      |
| L\$NUML | 000236   | #6-75      |
| L\$NXT  | 000044   | #6-75      |
| L\$OPT  | 000062   | #6-75      |
| L\$OWN  | 000102   | #6-75      |
| L\$PAR  | 000072   | #6-75      |
| L\$PDV  | 000055   | #6-75      |
| L\$PFG  | 000041   | #6-75      |
| L\$PLT  | 000112   | #6-75      |
| L\$PRO  | 000075   | #6-75      |
| L\$PVC  | 000004   | #6-75      |
| L\$RET  | 000256   | #6-75      |
| L\$SCN  | 000020   | #6-75      |
| L\$SCR  | 000166   | #6-75      |
| L\$SER  | 000230   | #6-75      |
| L\$SLEN | 000074   | #6-75      |
| L\$SLT  | 000042   | #6-75      |
| L\$SNM  | 000046   | #6-75      |
| L\$STA  | 000100   | #6-75      |
| L\$TAD  | 000103   | #6-75      |
| L\$TH1  | 000130   | #6-75      |
| L\$TH2  | 000131   | #6-75      |
| L\$TH3  | 000132   | #6-75      |
| L\$TNM  | 000052   | #6-75      |
| L\$TPT  | 000050   | #6-75      |
| L\$TRB  | 000040   | #6-75      |
| L\$TYP  | 000063   | #6-75      |
| L\$UNT  | 000003   | #6-75      |
| L\$XCH  | 000232   | #6-75      |
| L\$XMT  | 000120   | #6-75      |
| M\$AC   | = *****  | 4-1        |
| R\$PRO  | = *****  | 6-75       |
| R\$RTR  | = *****  | 8-132      |
| S\$BAS  | = *****  | 8-132      |
| S\$BDG  | = *****  | 8-178      |
| \$DSW   | = *****  | 8-158      |

\*8-182

6-75      6-75

8-119      9-205

4-2      5-64      8-116  
 6-75

```

309      .SBTTL  Compute address in the COMM/EXEC databases
310
311      ;*-COASN-Compute address in the COMM/EXEC databases
312
313      This is a co-routine to compute the address of the system line
314      entry in the reverse mapping table and co-call it's caller.
315
316      -
317      Calling sequence:
318      JSR      R5,COASN
319
320      Inputs:
321      R2 = SLN and logical tributary # of <Line-id>
322
323      Outputs: (to co-routine caller)
324      R5 = Address of the system line entry in the reverse mapping table
325
326      COASN:: SAVRG  <R4>      ; Get a free register
327      MOV      R5,-(SP)      ; Save return address (for co-call)
328
329      MOV      R2,R5         ; Compute address in reverse mapping table
330      BIC      #^C<377>,R5   ; Isolate SLN
331      ASL      R5            ; Form word index
332      ADD      @LLCTA,R5     ; Compute address in reverse mapping table
333      MOV      (R5),R4       ; Get contents of table
334      BPL      10$           ; If PL, point-to-point connection
335
336      MOV      R2,R5         ; Must be a multi-point line
337      SWAB     R5            ; Isolate logical tributary #
338      BIC      #^C<77>,R5    ;
339      ADD      R4,R5         ; Compute address in reverse mapping table
340      ASL      R5            ;
341
342      10$ CALL    @ (SP)+      ; Co-call the caller
343
344      RESRG    <R4>          ; Restore the register
345      MOV      (SP)+,R5      ; Restore R5 stored by initial JSR
346      RETURN
  
```

|  |   |
|--|---|
| <pre> 325 000526 326 000530 010546 327 328 000532 010205 329 000534 042705 177400 330 000540 006305 331 000542 067705 000000G 332 000546 01150' 333 000550 100006 334 335 000552 010205 336 000554 000305 337 000556 042705 177700 338 000562 060405 339 000564 006305 340 341 000566 342 343 000570 344 000572 012605 345 000574       </pre> | <pre>       MOV      R2,R5       BIC      #^C&lt;377&gt;,R5       ASL      R5       ADD      @LLCTA,R5       MOV      (R5),R4       BPL      10\$        MOV      R2,R5       SWAB     R5       BIC      #^C&lt;77&gt;,R5       ADD      R4,R5       ASL      R5       </pre> |
|--|---|



```

851 002152 005205          INC      R5          ; Set non-multiplexor flag
852
853 002154          20$: CALL      CDTB         ; Convert controller number
854 002160 103431          BCS      50$         ; If CS, illegal character
855 002162 110167 000002G  MOV     R1,$PBLK+T$CTL ; Save controller number
856 002166 020004          CMP      R0,R4        ; At end of the string?
857 002170 103023          BHIS     40$         ; If HIS, yes ... look for <Line-id>
858
859 002172 005705          TST      R5          ; Is device a multiplexor?
860 002174 001011          BNE      30$         ; If NE, no
861
862 002176 012703 000056    MOV     #'..R3      ; Set up terminating character
863 002202          CALL      CDTB         ; Convert line number
864 002206 103416          BCS      50$         ; If CS, illegal character
865 002210 110167 000003G  MOV     R1,$PBLK+T$LIN ; Save line number
866 002214 020004          CMP      R0,R4        ; At end of the string?
867 002216 103010          BHIS     40$         ; If HIS, yes ... look for <Line-id>
868
869 002220 005003          30$: CLR      R3          ; No special terminator required
870 002222          CALL      CDTB         ; Convert tributary #
871 002226 103406          BCS      50$         ; If CS, illegal character
872 002230 110167 000011G  MOV     R1,$PBLK+T$SLN+1 ; Save logical tributary #
873 002234 020004          CMP      R0,R4        ; At end of the string?
874 002236 103402          BLO      50$         ; If LO, no ... error (C-bit set)
875
876 002240          40$: CALL      FNDLA        ; Search databases for <Line-id>
877
878 002244          50$: .IF DF      R$$MPL      ; Restore mapping
879          RESMAP
880          .ENDC
881
882 002244          RFSRG     <R5,R4,R3>        ; Restore registers
883 002252          RETURN

```

## SYMBOL CROSS REFERENCE

CREF    04.00

## SYMBOL    VALUE      REFERENCES

|         |          |       |          |          |         |          |         |        |                    |
|---------|----------|-------|----------|----------|---------|----------|---------|--------|--------------------|
| SF.ENA  | = 000100 | #5-50 |          |          |         |          |         |        |                    |
| SF.LPB  | = 000004 | #5-50 | 17-669   |          |         |          |         |        |                    |
| SF.MFL  | = 000040 | #5-50 |          |          |         |          |         |        |                    |
| SF.PAC  | = 000020 | #5-50 |          |          |         |          |         |        |                    |
| SF.REA  | = 000010 | #5-50 |          |          |         |          |         |        |                    |
| SF.SER  | = 000001 | #5-50 |          |          |         |          |         |        |                    |
| SF.SVC  | = 000002 | #5-50 |          |          |         |          |         |        |                    |
| SF.UNL  | = 000040 | #5-50 |          |          |         |          |         |        |                    |
| SLTMA   | = *****  | GX    | 18-734   | 27-1091  |         |          |         |        |                    |
| SLTNM   | = *****  | GX    | 18-733   | 18-754   |         |          |         |        |                    |
| STACT   | 000020   | RG    | #6-90    |          |         |          |         |        |                    |
| STLNA   | 002464   | RG    | #25-1031 |          |         |          |         |        |                    |
| STLNK   | 002460   | RG    | #25-1029 |          |         |          |         |        |                    |
| STNMST  | 002516   | RG    | 6-107    | #26-1054 |         |          |         |        |                    |
| STSLT   | 002602   | RG    | 6-91     | 12-374   | 26-1055 | #27-1088 |         |        |                    |
| S.COST  | 000001   |       | #5-50    |          |         |          |         |        |                    |
| S.FLG   | 000000   |       | #5-50    |          |         |          |         |        |                    |
| S.LEN   | 000004   |       | #5-50    |          |         |          |         |        |                    |
| S.NMST  | 000002   |       | #5-50    | 15-531   | 18-770  | 26-1070  |         |        |                    |
| S.OWNR  | 000003   |       | #5-50    |          |         |          |         |        |                    |
| T\$ASN  | 000016   |       | *15-532  | 16-622   | *18-789 | 23-988   |         |        |                    |
| T\$CTL  | 000002   |       | 18-740   | *19-825  | *19-855 |          |         |        |                    |
| T\$DDM  | 000014   |       | *18-750  |          |         |          |         |        |                    |
| T\$FLG  | 000006   |       | 16-603   | *18-726  |         |          |         |        |                    |
| T\$KRB  | 000020   |       | *18-751  |          |         |          |         |        |                    |
| T\$LIN  | 000003   |       | 18-742   | *19-865  |         |          |         |        |                    |
| T\$NAM  | 000000   |       | 18-718   | *19-824  | *19-838 |          |         |        |                    |
| T\$NMST | 000022   |       | *15-531  | *18-752  | *18-770 |          |         |        |                    |
| T\$NSTA | 000005   |       | 17-664   | *18-758  |         |          |         |        |                    |
| T\$PDV  | 000007   |       | *18-722  | 18-738   |         |          |         |        |                    |
| T\$SFLG | 000004   |       | *15-530  | 17-669   | *18-769 | *19-826  |         |        |                    |
| T\$SLF  | 000012   |       | 16-605   | 16-607   | 17-662  | 17-667   | *18-749 | 18-775 |                    |
| T\$SLN  | 000010   |       | 15-511   | *15-526  | *18-756 | 18-762   | 18-773  | 18-785 | *19-827    *19-872 |
| T\$ST3  | = *****  | GX    | 16-568   |          |         |          |         |        |                    |
| T3.PRIV | = *****  | GX    | 16-568   |          |         |          |         |        |                    |
| X\$MBCB | = *****  |       | 5-48     | 5-48     |         |          |         |        |                    |
| ZF.CDU  | = 001000 |       | #5-48    |          |         |          |         |        |                    |
| ZF.DDM  | = 000001 |       | #5-48    | 16-603   | 18-730  |          |         |        |                    |
| ZF.DIA  | = 004000 |       | #5-48    |          |         |          |         |        |                    |
| ZF.DLC  | = 000002 |       | #5-48    |          |         |          |         |        |                    |
| ZF.DVP  | = 100000 |       | #5-48    |          |         |          |         |        |                    |
| ZF.INI  | = 040000 |       | #5-48    |          |         |          |         |        |                    |
| ZF.KMX  | = 000020 |       | #5-48    |          |         |          |         |        |                    |
| ZF.LLC  | = 000004 |       | #5-48    | 18-728   | 20-910  |          |         |        |                    |
| ZF.LMC  | = 000100 |       | #5-48    |          |         |          |         |        |                    |
| ZF.MAN  | = 020000 |       | #5-48    |          |         |          |         |        |                    |
| ZF.MFL  | = 000010 |       | #5-48    |          |         |          |         |        |                    |
| ZF.MTM  | = 000400 |       | #5-48    |          |         |          |         |        |                    |
| ZF.MUX  | = 000040 |       | #5-48    | 19-847   |         |          |         |        |                    |
| ZF.PSE  | = 002000 |       | #5-48    |          |         |          |         |        |                    |
| ZF.SLI  | = 010000 |       | #5-48    |          |         |          |         |        |                    |
| ZF.TIM  | = 000200 |       | #5-48    |          |         |          |         |        |                    |

```
230          BEQ      641$          ;BR IF NO
231          BIC      #T2.STP+2!T2.STP,T.ST2(R0) ;CLEAR TASK STOP BIT (DON'T CAL
232                                     ;EXRON - IT UNCONDITIONALLY SETS PRIORITY
233                                     ; TO PRO
234          BR       645$          ; AND EXIT
235
236          .ENDC
237          .ENDC
238
239 000320      641$: CALL      @EXRON          ; ...
240
241 000324 000241      645$: CLC                      ; INDICATE SUCCESS
242
243 000326      RESRG   <R4,R1,R0>          ; AND MAPPING
244
245 000334 006016      65$: ROR      (SP)          ; SAVE RETURN STATUS
246
247 000336 012703 000422'      70$: MOV      #ZERTAB,R3          ; POINT TO COUNTER CONTROL TABLE
248 000342 012302          MOV      (R3)+,R2          ; GET NEXT TABLE ENTRY
249 000344 001423          BEQ      100$          ; IF EQ, END OF TABLE
250 000346 020002          CMP      R0,R2          ; SHOULD WE ZERO THE COUNTERS FOR THIS EVENT?
251 000350 001374          BNE      70$          ; IF NE, NO ... KEEP LOOKING
252
253 000352 132711 000001      BITB     #EV.CCB,(R1)          ; DO WE HAVE A CCB WITH THIS EVENT?
254 000356 001402          BEQ      80$          ; IF EQ, NO
255 000360 016404 000016      MOV      C.BUF+2(R4),R4          ; POINT TO BUFFER
256
257 000364 116102 000001      80$: MOVB     1(R1),R2          ; GET # OF BYTES TO ZERO
258 000370 042702 177400      BIC      #^C<377>,R2          ;
259 000374 001407          BEQ      100$          ; IF EQ, NONE
260
261 000376          SAVRG   <R4>          ; SAVE POINTER
262 000400 105024      90$: CLR      (R4)+          ; ZERO THE COUNTER BLOCK
263 000402          SOB     R2,90$          ;
264 000406          RESRG   <R4>          ; RECOVER POINTER
265 000410 017714 177424      MOV      @ZTIM2,(R4)          ; FILL IN TIME SINCE MIDNIGHT
266
267 000414 000241      100$: CLC                      ; RESTORE STATUS
268 000416 006116      ROL      (SP)          ; ...
269 000420      110$: RETURN
270
271          ;+
272          ; COUNTER CONTROL TABLE
273          ; -
274
275 000422      ZERTAB: EVENT 0,8.          ; EVENT 0.8 - AUTOMATIC COUNTERS
276 000424          ENDEVENT
```

DLXCHR

J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D

DLXLIN

[illegible]

DLXSUB

|   |    |
|---|----|
| M | 14 |
| N | 14 |
| B | 15 |
| C | 15 |
| D | 15 |
| E | 15 |
| F | 15 |
| G | 15 |
| H | 15 |
| I | 15 |
| J | 15 |
| K | 15 |
| L | 15 |
| M | 15 |
| N | 15 |
| B | 16 |
| C | 16 |
| D | 16 |
| E | 16 |
| F | 16 |
| G | 16 |
| H | 16 |
| I | 16 |
| J | 16 |
| K | 16 |
| L | 16 |
| M | 16 |

EVL

## SYMBOL CROSS REFERENCE

CREF 04.00

## SYMBOL VALUE REFERENCES

|          |            |        |         |         |         |         |         |
|----------|------------|--------|---------|---------|---------|---------|---------|
| E\$MOD   | 000012     | #5-61  |         |         |         |         |         |
| E\$NOD   | 000010     | #5-61  |         |         |         |         |         |
| E\$PORT  | 000014     | #5-61  |         |         |         |         |         |
| E\$PRM   | 000002     | #5-61  |         |         |         |         |         |
| E\$STAT  | 000006     | #5-61  |         |         |         |         |         |
| E\$TCB   | 000004     | #5-61  | 9-143   | 9-224   | 10-305  | *10-311 | 11-374  |
| E.CTL    | 000020     | #5-61  | 9-152   | 9-169   | 9-171   | 9-177   | 9-181   |
| E.DATA   | 000046     | #5-61  | 9-187   | 9-189   | 9-194   |         | *13-471 |
| E.EVT    | 000002     | #5-61  | *9-150  |         |         |         |         |
| E.LCN    | 000042     | #5-61  |         |         |         |         |         |
| E.LEN    | 000216     | #5-61  | 9-187   | 9-189   |         |         |         |
| E.LIN    | 000024     | #5-61  | *9-173  | 12-396  | 12-40i  | *12-404 | *13-457 |
| E.LNK    | 000000     | #5-61  |         |         |         |         | *13-470 |
| E.MOD    | 000036     | #5-61  |         |         |         |         |         |
| E.NOD    | 000034     | #5-61  |         |         |         |         |         |
| E.PDV    | 000021     | #5-61  |         |         |         |         |         |
| E.PORT   | 000040     | #5-61  | 13-429  | 13-451  | 13-468  |         |         |
| E.PRM    | 000026     | #5-61  |         |         |         |         |         |
| E.PVC    | 000044     | #5-61  | *13-422 | *13-446 |         |         |         |
| E.SIZ    | 000022     | #5-61  | *9-190  |         |         |         |         |
| E.TIME   | 000004     | #5-61  | 14-493  |         |         |         |         |
| FR\$BCC  | = 000007   | #5-61  |         |         |         |         |         |
| FR\$CCF  | = 000001   | #5-61  |         |         |         |         |         |
| FR\$CDF  | = 000002   | #5-61  |         |         |         |         |         |
| FR\$DAO  | = 000011   | #5-61  |         |         |         |         |         |
| FR\$EXC  | = 000000   | #5-61  |         |         |         |         |         |
| FR\$FRM  | = 000010   | #5-61  |         |         |         |         |         |
| FR\$FTL  | = 000005   | #5-61  |         |         |         |         |         |
| FR\$OPN  | = 000004   | #5-61  |         |         |         |         |         |
| FR\$REFD | = 000006   | #5-61  |         |         |         |         |         |
| FR\$SBU  | = 000012   | #5-61  |         |         |         |         |         |
| FR\$SHO  | = 000003   | #5-61  |         |         |         |         |         |
| FR\$UBU  | = 000013   | #5-61  |         |         |         |         |         |
| FR\$UPT  | = 000014   | #5-61  |         |         |         |         |         |
| H\$PTB   | 000020     | 13-426 |         |         |         |         |         |
| H\$PVC   | 000006     | 13-444 |         |         |         |         |         |
| INIFLG   | 000530 R   | 10-291 | *10-303 | #10-317 |         |         |         |
| I\$AS    | = *****    | 5-62   |         |         |         |         |         |
| KISAR6   | = ***** GX | 8-97   | *9-174  | 13-421  | *13-462 | *13-473 |         |
| KSAR6    | = 000020 R | #8-97  |         |         |         |         |         |
| LF.ACT   | = 100000   | #5-60  |         |         |         |         |         |
| LF.BRO   | = 000400   | #5-60  |         |         |         |         |         |
| LF.BWT   | = 000007   | #5-60  |         |         |         |         |         |
| LF.ENA   | = 002000   | #5-60  |         |         |         |         |         |
| LF.LPB   | = 001000   | #5-60  |         |         |         |         |         |
| LF.MDC   | = 000100   | #5-60  |         |         |         |         |         |
| LF.MFL   | = 004000   | #5-60  |         |         |         |         |         |
| LF.MTP   | = 000020   | #5-60  |         |         |         |         |         |
| LF.PAC   | = 000200   | #5-60  |         |         |         |         |         |
| LF.RDY   | = 040000   | #5-60  |         |         |         |         |         |
| LF.REA   | = 100000   | #5-60  |         |         |         |         |         |
| LF.SER   | = 000040   | #5-60  |         |         |         |         |         |

```

180 000042 016702 140000'      MOV    .BASEB,R2      ; SAVE PNTR TO NEXT HOLE IN R2
181 000046 060577 177726      ADD     R5,AKS6      ; ADJUST BEGINNING OF HOLE
182 000052 010267 140000'      MOV     R2,.BASEB    ; SET NEXT HOLE PNTR & HOLE SIZE AT
183 000056 010067 140002'      MOV     R0,.BASEB+2  ; BASE OF ADJUSTED (SHORTENED) HOLE.
184 000062 060504      ADD     R5,R4      ; ADJUST HOLE POINTER IN TPD ENTRY
185 000064 000450      BR       60$      ; RETURN WITH R5=ADDRESS
186
187 000066 017746 177706      50$: MOV    AKS6,-(SP)    ; SET BIAS OF ALLOCATED SPACE ON STACK.
188 000072 016704 140000'      MOV     .BASEB,R4    ; SET HOLE POINTER IN TPD ENTRY TO POINT
189 000076 000443      BR       60$      ; TO NEXT HOLE (IF ANY), AND RETURN WITH R5 = ADDRESS
190
191      ; SPACE NOT FOUND IN FIRST HOLE -- CHECK REMAINING HOLES (IF ANY)
192
193 000100 017701 177674      40$: MOV    AKS6,R1      ; SAVE HOLE ADR IN R1 AS PREVIOUS HOLE PNTR
194 000104 016777 140000' 177666 MOV    .BASEB,AKS6    ; AND ADVANCE VIRTUAL BIAS TO NEXT HOLE.
195      ; IS THERE A NEXT HOLE?
196 000112 000261      SEC             ; ASSUME NOT
197 000114 001436      BEQ     70$      ; NO -- RETURN WITH CARRY SET
198 000116 016700 140002'      MOV     .BASEB+2,R0    ; SET R0 TO HOLE SIZE, ADJUSTED AS IF REQUIRED
199 000122 160500      SUB     R5,R0      ; SPACE HAS BEEN TAKEN FROM HOLE.
200      ; IS SUFFICIENT SPACE AVAILABLE?
201 000124 103765      BLO     40$      ; NO -- CHECK NEXT HOLE
202      ; YES -- IS ANY SPACE LEFT IN HOLE?
203 000126 001417      BEQ     50$      ; NO -- ALLOCATE MEMORY & DELETE HOLE
204      ; YES -- ALLOCATE MEMORY & ADJUST HOLE SIZE
205
206 000130 017746 177644      ;
207 000134 016702 140000'      MOV     AKS6,-(SP)    ; SET BIAS OF ALLOCATED SPACE ON STACK
208 000140 060577 177634      MOV     .BASEB,R2      ; SAVE PNTR TO NEXT HOLE IN R2
209 000144 010267 140000'      ADD     R5,AKS6      ; ADJUST BEGINNING OF HOLE
210 000150 010067 140002'      MOV     R2,.BASEB    ; SET NEXT HOLE PNTR & HOLE SIZE AT
211 000154 010177 177620      MOV     R0,.BASEB+2  ; BASE OF ADJUSTED (SHORTENED) HOLE.
212 000160 060567 140000'      MOV     R1,AKS6      ; ADJUST POINTER IN PREVIOUS HOLE.
213 000164 000410      ADD     R5,.BASEB
214      BR       60$      ; RETURN WITH R5=ADDRESS
215
216 000166 017746 177606      50$: MOV    AKS6,-(SP)    ; SET BIAS OF ALLOCATED SPACE ON STACK.
217 000172 016702 140000'      MOV     .BASEB,R2    ; SET POINTER TO NEXT HOLE (IF ANY) IN
218 000176 010177 177576      MOV     R1,AKS6      ; PREVIOUS HOLE.
219 000202 010267 140000'      MOV     R2,.BASEB
220      ; RETURN WITH R5=ADDRESS OF ALLOCATED SPACE
221 000206 000271      60$: CLC             ; COMMON CODE TO RETURN WITH R5=ADDRESS
222 000210 012605      MOV     (SP)+,R5    ; INDICATE SUCCESS
223 000212      70$:      ; GET ADDRESS FROM TOP OF STACK
224
225 000212 012677 177562      MOV     (SP)+,AKS6    ; RESTORE MAPPING REGISTER
226 000216      RETURN
227

```

CEALL - ALLOCATE BYTES OF POOL MACRO V05.03b Saturday 29-Jun-85 00:01 Page 8-1

## Symbol table

|                  |                   |                  |                   |                   |
|------------------|-------------------|------------------|-------------------|-------------------|
| ALOCB = ***** GX | D\$\$YNC= 000000  | K\$\$TPS= 000074 | N\$\$LDV= 000001  | S\$\$WRG= 000000  |
| ALOC1 = ***** GX | D\$\$YNM= 000000  | LD\$LP = 000000  | N\$\$MCP= 000001  | S\$\$YSZ= 007600  |
| A\$\$CHK= 000000 | EX\$IZ = ***** GX | L\$\$ASG= 000000 | N\$\$MML= 000001  | T\$\$KMG= 000000  |
| A\$\$CPS= 000000 | E\$\$XPR= 000000  | L\$\$DRV= 000000 | N\$\$MOV= 000010  | T\$\$MIN= 000000  |
| A\$\$PRI= 000000 | F\$\$LVL= 000001  | L\$\$P11= 000001 | N\$\$NCT= 000001  | V\$\$CTR= 001000  |
| A\$\$TRP= 000000 | G\$\$TTP= 000000  | L\$\$11R= 000000 | N\$\$PEM= 000001  | X\$\$DBT= 000000  |
| CEAVL = ***** GX | G\$\$TSS= 000000  | M\$\$CRB= 000124 | P\$\$P45= 000000  | \$ALL16 000000RG  |
| CRAVL = ***** GX | G\$\$TTP= 000000  | M\$\$CRX= 000000 | P\$\$WRD= 000000  | \$ALL18 000150RG  |
| C\$\$CKP= 000000 | G\$\$WRD= 000000  | M\$\$FCS= 000000 | QBIAS = ***** GX  | \$ALN18 000120RG  |
| C\$\$ORE= 000400 | HEADR = ***** GX  | M\$\$MGE= 000000 | Q\$TRT = ***** GX | \$AMEM = ***** GX |
| C\$\$RSH= 177564 | I\$\$RAR= 000000  | M\$\$NET= 000000 | Q\$\$OPT= 000010  | \$DEA16 000060RG  |
| DEAC1 = ***** GX | I\$\$RDN= 000000  | M\$\$GVR= 000000 | R\$\$DER= 000000  | \$DEA18 000220RG  |
| D\$\$BUG= 177514 | K\$\$CNT= 177546  | NBIAS = ***** GX | R\$\$K11= 000001  | \$DECEX 000064RG  |
| D\$\$ISK= 000000 | K\$\$CSR= 177546  | N\$\$ACC= 000001 | R\$\$SND= 000000  | \$DMEM = ***** GX |
| D\$\$L11= 000001 | K\$\$LDC= 000000  | N\$\$BUF= 000001 | R\$\$11M= 000000  |                   |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
 000276 001 (RW,I,LCL,REL,CON)

Errors detected: 0

## \*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 8936 Words ( 35 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:07.39

SY:CEALL.V2,[132,134]CEALL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CEALL

CFGCNT      CREATED BY MACRO ON 29-JUN-85 AT 00:01      PAGE 1      4

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL   | VALUE      | REFERENCES  |
|----------|------------|---|
| CERR     | = ***** GX | 6-77      6-78      6-79      6-80      6-81                    |
| CFGBF    | = ***** GX | 7-108   |
| CFGSZ    | = ***** GX | 7-107   |
| CFLIN    | = ***** GX | 6-77      6-78      6-79      6-80      6-81                    |
| CNTDF    | 000000 R   | 7-104   |
| CNTKW    | 000000 RG  | 7-106      #8-123   |
| CNTST    | 000000 RG  | #8-123  |
| CTL      | = ***** GX | 9-170   |
| C.CSR    | 000164 R   | #9-194  |
| C.CTL    | 000052 R   | #9-170  |
| C.PRI    | 000240 R   | #9-216  |
| C.VECT   | 000074 R   | #9-177  |
| FMT8     | = ***** GX | 5-78  |
| FMT8A    | = ***** GX | 5-81  |
| FMT8B    | = ***** GX | 6-77      6-79      6-80  |
| FM.8     | = 000000   | #6-78   |
| FM.8A    | = 000000   | #6-81   |
| FM.8B    | = 000000   | #6-77      #6-79      #6-80                                     |
| GETCSR   | = ***** GX | 9-196   |
| J\$AS    | = *****    | 9-224   |
| MS.CSR   | = ***** GX | 9-194      9-209  |
| MS.PRI   | = ***** GX | 9-216      9-226  |
| MS.VCT   | = ***** GX | 9-177      9-186  |
| REP8A    | = ***** GX | 6-81  |
| RTSPC    | = ***** GX | 6-77      6-78      6-79      6-80                              |
| R\$EIS   | = *****    | 9-224   |
| R\$EMPL  | = *****    | 6-82      8-142      9-199      9-233                           |
| R\$11D   | = *****    | 9-224   |
| SYNERR   | = ***** GX | *7-109      *7-112      *9-172                                  |
| S\$BAS   | = *****    | 6-77      6-78      6-79      6-80      6-81                    |
| \$ALPHA  | = 000022   | #8-123  |
| \$ANY    | = 000020   | #8-123  |
| \$BLANK  | = 000006   | #8-123  |
| \$DIGIT  | = 000024   | #8-123  |
| \$DNUMB  | = 000014   | #8-123  |
| \$EOS    | = 000012   | #8-123  |
| \$ERRZ2  | 000150 R   | #6-81      9-212  |
| \$ERR1C  | 000000 R   | #6-77      9-188  |
| \$ERR1D  | 000032 R   | #6-78      9-189  |
| \$ERR1E  | 000070 R   | #6-79      9-211  |
| \$ERR1F  | 000114 R   | #6-80      9-228  |
| \$ERR1T  | = ***** GX | 7-119   |
| \$EXIT   | = 000000   | #8-123  |
| \$FAIL   | = 177777   | #8-123  |
| \$GPRM   | = *****    | 8-123   |
| \$LAMBDA | = 000000   | #8-123  |
| \$MISS   | = ***** GX | 9-177      *9-186      9-194      *9-209      9-216      *9-226 |
| \$MXVCT  | = ***** GX | 9-183   |
| \$NLCSR  | = ***** GX | 9-206   |
| \$NUMBER | = 000002   | #8-123  |
| \$GCNT   | 000000 RG  | #7-104  |



CFGDDM      CREATED BY    MACRO    ON 29-JUN-85 AT 00:02      PAGE 2      B 5

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES  |
|------------|---|
| CALL       | 6-79  |
| DBGTP\$    | #7-92      #7-98      #7-112      #7-117      #7-132      #7-133      #7-134      #7-137      #7-138      #7-141  |
|            | #7-142      #7-148  |
| LMSG\$R    | #5-57      6-88   |
| ISTAT\$    | #5-57      7-92   |
| MTRANS     | #7-92   |
| RETURN     | 6-83      8-161   |
| STAT\$     | #5-57      7-96      #7-99      #7-104      #7-106      #7-108      #7-110      #7-113      #7-115      #7-118    |
|            | 7-130      #7-135      #7-139      #7-146      #7-150   |
| TRANS      | #5-57      #7-97      #7-98      7-103      #7-105      #7-107      #7-109      #7-111      #7-112      #7-114    |
|            | #7-116      #7-117      #7-119      7-131      #7-132      #7-133      #7-134      #7-136      #7-137      #7-138 |
|            | #7-140      #7-141      #7-142      #7-147      #7-148  |

```

000302 016700 000000G DTENET: MOV ,PSTCN,R0 ; GET SIZE OF SUPPLIED NAME
000306 000306 020027 000006 CMP R0,#DTNMMX ; BIGGER THAN MAXIMUM?
000312 101010 BHI 101$ ; BR IF YES, ERROR
000314 016701 000000G MOV ,PSTPT,R1 ; POINT TO NAME
000320 012702 000027' MOV #DTN:NM,P2 ; POINT TO LOCAL COPY
000324 112122 10$: MOV B (R1)+,(R2)+ ; COPY NAME
000326 005300 DEC R0 ; MORE TO COPY?
000330 003375 BGT 10$ ; BR IF YES
000332 000403 BR DTEEND ; ELSE MERGE INTO END PROCESSING
;
; ERRORS
000334 101$: MSG$R 00 ; ILLEGAL NETWORK NAME PARAMETER

```

FILEIDCFGFEA

B 7

|          |          |          |          |          |          |    |
|----------|----------|----------|----------|----------|----------|----|
| CCCCCCCC | FFFFFFFF | GGGGGGGG | FFFFFFFF | EEEEEEEE | AAAAAA   |    |
| CCCCCCCC | FFFFFFFF | GGGGGGGG | FFFFFFFF | EEEEEEEE | AAAAAA   |    |
| CC       | FF       | GG       | FF       | EE       | AA       | AA |
| CC       | FF       | GG       | FF       | EE       | AA       | AA |
| CC       | FF       | GG       | FF       | EE       | AA       | AA |
| CC       | FF       | GG       | FF       | EE       | AA       | AA |
| CC       | FFFFFFFF | GG       | FFFFFFFF | EEEEEEEE | AA       | AA |
| CC       | FFFFFFFF | GG       | FFFFFFFF | EEEEEEEE | AA       | AA |
| CC       | FF       | GG       | GGGGGG   | EE       | AAAAAAAA |    |
| CC       | FF       | GG       | GGGGGG   | EE       | AAAAAAAA |    |
| CC       | FF       | GG       | GG       | EE       | AA       | AA |
| CC       | FF       | GG       | GG       | EE       | AA       | AA |
| CC       | CCCCCCCC | GGGGGG   | FF       | EEEEEEEE | AA       | AA |
| CC       | CCCCCCCC | GGGGGG   | FF       | EEEEEEEE | AA       | AA |

|          |        |          |
|----------|--------|----------|
| LL       | SSSSSS | TTTTTTTT |
| LL       | SSSSSS | TTTTTTTT |
| LL       | SS     | TT       |
| LL       | SS     | TT       |
| LL       | SS     | TT       |
| LL       | SS     | TT       |
| LL       | SSSSSS | TT       |
| LL       | SSSSSS | TT       |
| LL       | SS     | TT       |
| LL       | SS     | TT       |
| LL       | SS     | TT       |
| LL       | SS     | TT       |
| LLLLLLLL | SSSSSS | TT       |
| LLLLLLLL | SSSSSS | TT       |

C 7

```

54      .SBTTL  MACRO DEFINITIONS
55
56      ;
57      ; LIBRARY MACROS
58      ;
59      .MACALL  EMSG$R,SAVRG,RESRG,RETC,PVCDF$,XCBDF$
60      .MACALL  PHBDF$,NILER$,STAT$,TRANS$,ISTAT$
61      .MACALL  SLTDF$,XPDDB$,CTDRF$,DTEDF$,PLBDF$
62
63      ; LIBRARY SYMBOLS
64      ;
65      000000      DTEDF$      ; LOCAL DTE DESCRIPTOR OFFSETS
66      000000      CTDRF$      ; TRANSPORT COUNTER BLOCK OFFSETS
67      000000      XPDDB$      ; XPT'S DDB OFFSETS
68      000000      PHBDF$      ; PSI HOME BLOCK OFFSETS
69      000000      PVCDF$      ; PVC NAME BLOCK OFFSETS
70      000000      SLTDF$      ; SLT OFFSETS
71      000000      XCBDF$      ; X25 CIRCUIT CONTROL BLOCKS
72      000000      PLBDF$      ; DEFINE PLB OFFSETS
73
74      ; LOCAL MACRO DEFINITIONS
75      ;
76      .MACRO  SAVMAP
77      MOV     @KSAR5,-(SP)      ; SAVE APR 5
78      .ENDM
79
80      .MACRO  RESMAP
81      MOV     (SP)+,@KSAR5      ; RESTORE APR5
82      .ENDM
83
84      ; SET UP APR5 MAPPING
85      ;
86      .MACRO  MAP      BIAS
87      MOV     BIAS,@KSAR5      ; SET UP APR5 MAPPING
88      .ENDM
89
90      ; SET UP APR5 BIAS
91      ;
92      .MACRO  BIAS      REG
93      BIC     #160000,REG      ; CLEAR CURRENT MAPPING BITS
94      BIS     #120000,REG      ; SET UP APR5 MAPPING
95      .ENDM
  
```

```

548 .SBTTL FNDPRT - FIND FREE PORT ENTRY
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
001172
001176 017700 000000G
001202 016067 000022 000004'
001210 016046 000020
001214
001220 012600
001222 005001
001224 005720
001226 005201
001230 005720
001232 001006
001234 005740
001236 010067 000006'
001242 110167 000034'
001246 000404
001250 020167 000004'
001254 1C3764
001256 000261
001260
001264
FNDPRT: SAVMAP
MOV @PSIPT,R0
MOV H$NPT(R0),PORTNO
MOV H$PTB(R0),-(SP)
CALL $CEACX
MOV (SP)+,R0
CLR R1
TST (R0)+
10$: INC R1
TST (R0)+
BNE 15$
TST -(R0)
MOV R0,PRTADD
MOVB R1,PVCPRT
BR 20$
15$: CMP R1,PORTNO
BLO 10$
SEC
20$: RESMAP
RETURN

```

CFGPVC CREATED BY MACRO ON 29-JUN-85 AT 00:04 PAGE 3 B 10  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|---|
| L.LEN   | = 000022   | #5-70   |
| L.MPF   | 000022     | #5-70   |
| L.NMST  | 000020     | #5-70   |
| L.NSTA  | 000014     | #5-70   |
| L.OWNR  | 000021     | #5-70   |
| L.UNIT  | 000013     | #5-70   |
| NS.TLC  | 000100     | 21-804 21-805 21-806                              |
| NS\$VCT | = *****    | 13-423 13-441 13-450 15-544 20-720                |
| PCVKW   | 000000 RG  | 9-189 #10-208                                     |
| PDVID   | = ***** GX | 20-718  |
| PDVNM   | = ***** GX | 21-793  |
| PDVTA   | = ***** GX | 20-723 21-792                                     |
| PF.DLM  | = 000001   | #7-142 13-411 13-419 13-443 13-451 14-496         |
| PORTNO  | 000004 R   | #7-130 *16-567 16-580                             |
| PRTADD  | 000006 R   | #7-131 14-488 *16-577                             |
| PSIPT   | = ***** GX | 13-453 14-484 15-522 16-566                       |
| PVCCTM  | 000012 R   | #7-133 *11-342 *7-614                             |
| PVCDF   | 000000 R   | 9-187   |
| PVCFLG  | 000035 R   | #7-144 *11-314 *12-396 13-436                     |
| PVCLCN  | 000010 R   | #7-132 *11-330 13-437 15-532 17-617 18-657 18-673 |
| PVCNAM  | 000036 R   | #7-145 11-301 13-432 15-537 20-711                |
| PVCOWN  | 000020 R   | #7-136 *11-313 *12-382 14-492                     |
| PVCPRT  | 000034 R   | #7-143 *16-578 17-616 18-674                      |
| PVCPSZ  | 000014 R   | #7-134 *11-311 *12-355 17-613                     |
| PVCSZ   | 000000 RG  | #10-208   |
| PVCT    | 000220 R   | #11-340   |
| PVCTI   | 001726 R   | #20-737   |
| PVCL\$7 | 000016 R   | #7-135 *11-312 *12-367 17-612                     |
| PVDLM   | 001630 R   | #20-711   |
| PVEND   | 000400 R   | #13-407   |
| PVFG    | 000346 R   | #12-392   |
| PVLCN   | 000162 R   | #11-324   |
| PVNAM   | 000056 R   | #11-298   |
| PVNAMX  | = 000006   | #6-103 11-299 11-302 13-431 15-536                |
| PVOWN   | 000310 R   | #12-377   |
| PVPS7   | 000244 R   | #12-353   |
| PVSTA   | 002000 R   | #20-749   |
| PVWSZ   | 000266 R   | #12-365   |
| PX\$DLM | = 000200   | 21-817  |
| RTSPC   | = ***** GX | 8-161 8-162 8-163 8-164 8-165 8-166 8-167 8-168   |
| R\$R0   | = 000002   | #6-108  |
| R\$R1   | = 000004   | #6-109  |
| R\$R2   | = 000006   | #6-110  |
| R\$R3   | = 000010   | #6-111  |
| R\$R4   | = 000012   | #6-112  |
| R\$R5   | = 000014   | #6-113  |
| SF.ACT  | = 000200   | #5-70   |
| SF.ENA  | = 000100   | #5-70   |
| SF.LPB  | = 000004   | #5-70   |
| SF.MFL  | = 000040   | #5-70   |
| SF.PAC  | = 000020   | #5-70   |
| SF.REA  | = 000010   | #5-70   |

```

188          .SBTTL ACTION ROUTINES
189          ;
190          ; CONTROLLER NUMBER (SLT$DF)
191          ;
192 000062 126767 000000G 000000G S.CTL: CMPB .PNUMB,CTL ; REJECT TRANSITION IF NO MATCH
193 000070 001404          BEQ 10$          ;
194 000072          REJ$          ;
195 000102          10$: RETURN          ;
196          ;
197          ; UNIT NUMBER (SLT$DF)
198          ;
199          ;
200 000104 126767 000000G 000000G S.UNT: CMPB .PNUMB,UNT ; REJECT TRANSITION IF NO MATCH
201 000112 001404          BEQ 10$          ;
202 000114          REJ$          ;
203 000124          10$: RETURN          ;
204          ;
205          ; COUNTER TIMER (SLT$DF)
206          ;
207 000126 005767 000000G          S.CTIM: TST .PNUMH          ; VALID COUNTER TIMER VALUE?
208 000132 001004          BNE 101$          ; BR IF NO
209 000134 016767 000000G 000000G      MOV .PNUMB,$CTIM          ; ELSE SAVE COUNTER TIMER VALUE
210 000142          RETURN          ;
211 000144          101$: MSG$R YR          ; ILLEGAL COUNTER TIMER VALUE
212          ;
213          ; HELLO/LISTEN TIMER
214          ;
215          ;
216 000152 005767 000000G          S.HTIM: TST .PNUMH          ; DOUBLE WORD VALUE
217 000156 001011          BNE 101$          ; IF YES - BRANCH
218 000160 016767 000000G 000000G      MOV .PNUMB,$$HTIM          ; STORE HELLO TIMER VALUE
219 000166 006367 000000G          ASL .PNUMB          ; MULTIPLY BY 2
220 000172 016767 000000G 000000G      MOV .PNUMB,$$LTIM          ; STORE LISTEN TIMER
221 000200          RETURN          ;
222          ;
223 000202          101$: MSG$R ZP          ; ILLEGAL HELLO TIMER VALUE
224          ;
225          ; BROADCAST ROUTER ADJACENCIES
226          ;
227          ;
228          ;
229 000210 016767 000000G 000000G S.BRAD: MOV .PNUMB,.NBRA          ; STORE BROADCAST ROUTER ADJACENCIES VALUE
230 000216          RETURN          ; LIMIT CHECKED DURING MARK FOR LOAD SCAN
231          ;
232          ;
233          ; ROUTER PRIORITY
234          ;
235          ;
236 000220 022767 000177 000000G S.PPRI: CMP #127...PNUMB          ; LEGAL VALUE ?
237 000226 002404          BLT 101$          ; IF NO - BRANCH
238 000230 016767 000000G 000000G      MOV .PNUMB,.RPRI          ; STORE ROUTER PRIORITY
239 000236          RETURN          ;
240          ;
241 000240          101$: REJ$          ; ILLEGAL ROUTER PRIORITY VALUE
242 000250          RETURN          ;
243          ;
244          000001          .END

```

169 000270  
170 000270  
171 000270  
172  
173 000270

TRANS <','>,\$EXIT  
STATE\$ BITS2  
TRANS \$RAD50,BITS1  
STATE\$

CFG\$  
ERRC



```
135                                     .SBTTL  ERROR MESSAGES
136
137                                     .ENABL  LC
138 000056                             NTLR$ ,3D,8B,CFERR,RTSPC,,<packet size>
139 000102                             NTLR$ ,3E,8B,CFERR,RTSPC,,<window size>
140 000126                             NTLR$ ,3F,8B,CFERR,RTSPC,,<flags word>
141 000152                             NTLR$ ,3G,8B,CFERR,RTSPC,,<recall timer>
142 000200                             NTLR$ ,3H,8B,CFERR,RTSPC,,<retry timer>
143 000224                             NTLR$ ,3I,8B,CFERR,RTSPC,,<DTE address>
144 000250                             NTLR$ ,3J,8B,CFERR,RTSPC,,<owner>
145 000266                             NTLR$ ,3K,8,CFERR,RTSPC,,<SVL descriptor allocation failure>
146 000340                             NTLR$ ,3M,8,CFERR,RTSPC,,<DTE address missing>
147 000374                             NTLR$ ,3Z,8,CFERR,RTSPC,,<PSI not loaded>
148                                     .DSABL  LC
149
150 000000                             .PSECT
```

CFGSV0 - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 00:06 Page 17  
 STRNXT - STORE NEXT DIGIT

```

622                                     .SBTTL STRNXT - STORE NEXT DIGIT
623
624                                     ;+
625                                     ;
626                                     ; STRNXT - STORE NEXT DIGIT
627                                     ;
628                                     ; INPUTS:
629                                     ; .PCHAR - MATCHED CHARACTER FROM TPARS TRANSITION
630                                     ; NEXT - ADDRESS OF NEXT BYTE AVAILABLE FOR STORAGE
631                                     ;
632                                     ; OUTPUTS:
633                                     ; NEXT - ADDRESS OF NEXT AVAILABLE BYTE
634                                     ; COUNT - COUNTER
635                                     ; RO,R1 DESTROYED
636                                     ;
637 001520 116767 000000G 000054' STRNXT: MOVB .PCHAR,CVTBUF      ; MOVE DIGIT TO CONVERSION BUFFER
638 001526 012700 000054'      MOV #CVTBUF,R0      ; POINT TO CONVERSION BUFFER
639 001532      CALL $CMTB      ; CONVERT TO BINARY
640 001536 110177 000000'      MOVB R1,@NEXT      ; STORE BINARY DIGIT
641 001542 005267 000000'      INC NEXT          ; POINT TO NEXT AVAILABLE BYTE
642 001546 105267 000024'      INCB COUNT        ; INCREMENT DIGIT COUNT
643 001552      RETURN
644
645      000001      .END

```

CFGUNT - CONFIG FILE SCAN ACTION MACRO V05.03b Monday 15-Jul-85 18:56 Page 6  
 LOCAL DATA AND CONSTANT DEFINITIONS

```

62                                     .SBTTL  LOCAL DATA AND CONSTANT DEFINITIONS
63
64                                     ;
65                                     ; LOCAL DATA
66                                     ;
67                                     ;
68                                     ;
69                                     ; THIS TABLE IS INDEXED BY THE VALUE SPECIFIED IN THE HIGH BYTE
70                                     ; OF THE FIRST CHARACTERISTICS WORD. IT MAPS ONTO THE CORRECT
71                                     ; TRANSMIT/RECEIVE SPEED WHICH WAS SPECIFIED AT NETGEN
72                                     ;
73 000000 000062  SPEED: .WORD 50.
74 000002 000113      .WORD 75.
75 000004 000156      .WORD 110.
76 000006 000206      .WORD 134.
77 000010 000226      .WORD 150.
78 000012 000454      .WORD 300.
79 000014 001130      .WORD 600.
80 000016 002260      .WORD 1200.
81 000020 003410      .WORD 1800.
82 000022 003720      .WORD 2000.
83 000024 004540      .WORD 2400.
84 000026 007020      .WORD 3600.
85 000030 011300      .WORD 4800.
86 000032 016040      .WORD 7200.
87 000034 022600      .WORD 9600.
88 000036 022600      .WORD 9600.
89
90                                     ;
91                                     ; CONSTANT DEFINITIONS
92                                     ;
93 000031      MAXCST = 25. ; MAXIMUM COST ALLOWABLE
94
95                                     ;
96                                     ; CHARACTERISTICS WORD 1 PROTOCOL BIT SELECTIONS
97                                     ;
98 000001      C1.DCP = 1 ; DDCMP
99 000002      C1.BSY = 2 ; BISYNC
000003      C1.SDL = 3 ; SDLC/ADCCP/HDLC
000004      C1.X2 = 4 ; X.25

```

CFGU

SYMB

SYMB

CERR

CFGB

CFG

CFLI

CHER

CH.S

C1.B

C1.D

C1.S

C1.X

FL.F

FL.H

FL.K

FL.L

FMT8

FMT8

FMT8

FMT8

FMT8

GETC

LF.A

LF.E

LF.E

LF.L

LF.M

LF.M

LF.M

LF.M

LF.F

LF.F

LF.F

LF.S

LF.S

LF.S

LF.S

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

LN.C

CFGUNT      CREATED BY    MACRO    ON 15-JUL-85 AT 18:57      PAGE 1      **B 16**

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES  |
|--------|------------|---|
| CERR   | = ***** GX | 7-110      7-111      7-112      7-113      7-114      7-115      7-116 |
| CFGBF  | = ***** GX | 8-139   |
| CFGSZ  | = ***** GX | 8-138   |
| CFLIN  | = ***** GX | 7-110      7-111      7-112      7-113      7-114      7-115      7-116 |
| CHERR  | = 000254 R | 11-229      #11-246      11-252      11-268                             |
| CH.SYN | = 000004   | 11-306  |
| CI.BSY | = 000002   | #6-97   |
| CI.DCP | = 000001   | #6-96      12-287   |
| CI.SDL | = 000003   | #6-98   |
| CI.X25 | = 000004   | #6-99   |
| FL.FDX | = ***** GX | 11-235  |
| FL.HDX | = ***** GX | 11-238  |
| FL.KMX | = ***** GX | 13-342      13-372  |
| FL.LMC | = ***** GX | 13-339      13-370  |
| FMT8   | = ***** GX | 7-111      7-114      7-115   |
| FMT8A  | = ***** GX | 7-116   |
| FMT8B  | = ***** GX | 7-110      7-112      7-113   |
| FM.8   | = 000000   | #7-111      #7-114      #7-115  |
| FM.8A  | = 000000   | #7-116  |
| FM.8B  | = 000000   | #7-110      #7-112      #7-113  |
| GETCSR | = ***** GX | 13-354  |
| LF.ACT | = 100000   | #5-60   |
| LF.BRO | = 000400   | #5-60      13-392   |
| LF.BWT | = 000007   | #5-60   |
| LF.ENA | = 002000   | #5-60   |
| LF.LPB | = 001000   | #5-60   |
| LF.MDC | = 000100   | #5-60   |
| LF.MFL | = 004000   | #5-60   |
| LF.MTP | = 000020   | #5-60   |
| LF.PAC | = 000200   | #5-60   |
| LF.RDY | = 040000   | #5-60   |
| LF.REA | = 010000   | #5-60   |
| LF.SER | = 000040   | #5-60   |
| LF.TIM | = 000010   | #5-60   |
| LF.UNL | = 020000   | #5-60   |
| LF.X2P | = 000000   | #5-60   |
| LN.CLO | = 000000   | #5-60   |
| LN.DUM | = 000005   | #5-60   |
| LN.LOA | = 000004   | #5-60   |
| LN.LOO | = 000003   | #5-60   |
| LN.OAU | = 000003   | #5-60   |
| LN.OFF | = 000001   | #5-60   |
| LN.ON  | = 000000   | #5-60   |
| LN.OOP | = 000004   | #5-60   |
| LN.OPE | = 000001   | #5-60   |
| LN.REF | = 000002   | #5-60   |
| LN.SER | = 000002   | #5-60   |
| LN.STA | = 000017   | #5-60   |
| LN.SUB | = 000360   | #5-60   |
| LN.TRI | = 000006   | #5-60   |
| L.COST | = 000015   | #5-60      *13-397  |
| L.CTL  | = 000012   | #5-60   |

EVL CREATED BY MACRO ON 28-JUN-85 AT 22:58 PAGE 4 C 1  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL   | VALUE    | REFERENCES                |
|----------|----------|---------------------------|
| LF.TIM   | = 000010 | #5-60                     |
| LF.UNL   | = 020000 | #5-60                     |
| LF.X2P   | = 000000 | #5-60                     |
| LGDDB    | 000022 R | #8-98 10-301              |
| LGPDV    | 000024 R | #8-99 10-302              |
| LGSTT    | 000026 R | #8-100 10-309             |
| LN.CLO   | = 000000 | #5-60                     |
| LN.DUM   | = 000005 | #5-60                     |
| LN.LOA   | = 000004 | #5-60                     |
| LN.LOO   | = 000003 | #5-60                     |
| LN.OAU   | = 000003 | #5-60                     |
| LN.OFF   | = 000001 | #5-60                     |
| LN.ON    | = 000000 | #5-60                     |
| LN.OOP   | = 000004 | #5-60                     |
| LN.OPE   | = 000001 | #5-60                     |
| LN.REF   | = 000002 | #5-60                     |
| LN.SER   | = 000002 | #5-60                     |
| LN.STA   | = 000017 | #5-60                     |
| LN.SUB   | = 000360 | #5-60                     |
| LN.TRI   | = 000006 | #5-60                     |
| L\$S11   | = *****  | 9-199 9-223 11-338 11-368 |
| L.COST   | 000015   | #5-60                     |
| L.CTL    | 000012   | #5-60                     |
| L.CVA    | 177776   | #5-60                     |
| L.DDM    | 000002   | #5-60                     |
| L.DDS    | 000004   | #5-60                     |
| L.DLC    | 000003   | #5-60                     |
| L.DLM    | 000006   | #5-60                     |
| L.DLS    | 000010   | 13-462 13-463             |
| L.FLG    | 000000   | #5-60                     |
| L.KRBA   | 000016   | #5-60                     |
| L.LEN    | = 000022 | #5-60                     |
| L.MPF    | 000022   | #5-60                     |
| L.NMST   | 000020   | #5-60                     |
| L.NSTA   | 000014   | #5-60                     |
| L.OWNR   | 000021   | #5-60                     |
| L.UNT    | 000013   | #5-60                     |
| MAPCHN   | 000636 R | 9-179 #12-395             |
| MAPORT   | 000702 R | 9-183 #13-421             |
| MOS\$AC  | = 000016 | #5-61                     |
| MOS\$PR  | = 000012 | #5-61                     |
| MOS\$SV  | = 000014 | #5-61                     |
| MOS\$25A | = 000006 | #5-61                     |
| MOS\$25P | = 000002 | #5-61                     |
| MOS\$25S | = 000004 | #5-61                     |
| MOS\$29S | = 000010 | #5-61                     |
| M\$HIGH  | = 000003 | #5-61                     |
| M\$3'00  | = 000000 | #5-61                     |
| M\$3'01  | = 000001 | #5-61                     |
| M\$3'02  | = 000002 | #5-61                     |
| M\$3'05  | = 000003 | 5-61                      |
| NELOG    | 000042 R | 7-77 #9-143               |

.SBTTL \$DMEM - DEALLOCATE MEMORY FROM PARTITION

;\$DMEM - SUBROUTINE TO FREE A CONTIGUOUS SEGMENT OF REAL  
 MEMORY FROM A PARTITION.

INPUTS:

R3 ~ BASE ADDRESS OF SEGMENT TO BE RETURNED  
 R4 ~ CURRENT VALUE OF \$QBIAS  
 R5 ~ SIZE OF SEGMENT IN 32 WORD BLOCKS  
 TASK SWITCHING MUST BE DISABLED

OUTPUTS:

MEMORY IS RETURNED TO NTPOOL  
 R4 ~ NEW VALUE OF \$QBIAS  
 R0,R1,R2 -DESTROYED

CAUTION:

THIS ROUTINE CAN BE SUPPLIED WITH ANY ARGUMENTS AND IT WILL  
 SIMPLY ADD WHATEVER SEGMENT OF CORE THAT DEFINES TO THE FREE  
 SPACE LIST.  
 \$QBIAS MUST BE UPDATED IN IT'S GLOBAL STORAGE LOCATION  
 BEFORE TASK SWITCHING IS REENABLED.

\$DMEM::

```

MOV    KSAR6,KS6      ; COPY KISAR6 ADDRESS
MOV    @KS6,-(SP)      ; SAVE MAPPING REGISTER

MOV    R5,-(SP)        ; PUT LENGTH ON TOP OF STACK
MOV    KS6,R5
MOV    R3,(R5)         ; SET LOCATION 0 OF TASK SEGMENT HOLE TO
MOV    #BASEB+2,R2     ; VIRTUAL MPBIAS. PUT (T.HP) AS
MOV    (SP),(R2)       ; A FIRST (BUT NOT NECESSARILY
MOV    R4,-(R2)        ; FINAL) GUESS FOR TASK HOLE
                        ; POINTER. SET HOLE SIZE = TO
                        ; SEGMENT SIZE. R3,(SP)
                        ; ARE 1/64 THE
                        ; BASE ADDR. SEGMENT SIZE
                        ; RESPECTIVELY. T.HP IS INDEX
                        ; INTO TPD FOR 1ST HOLE
                        ; POINTER.

BEQ     20$,R4         ; IF TASK HOLE IS LOWER THAN
CMP     R3,R4         ; 1ST HOLE OR IF (T.HP) = 0,
                        ; THEN TASK HOLE BECOMES
                        ; 1ST HOLE.

BHI     30$,R4         ; NEW 1ST HOLE?

20$:    MOV    R3,R4   ; YES--REDEFINE 1ST HOLE POINTER.

30$:    MOV    R4,(R5) ; SET UTILITY ASR (KP.AR3) TO
                        ; 1ST HOLE AND BEGIN SCAN.
```

CEALL      CREATED BY MACRO ON 29-JUN-85 AT 00:01      PAGE 1      C 3

SYMBOL CROSS REFERENCE      CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                             |
|---------|------------|--|
| ALOCB   | = ***** GX | 5-82                                   |
| ALOC1   | = ***** GX | 5-77                                   |
| CEAVL   | = ***** GX | 5-73      6-112                        |
| CRAVL   | = ***** GX | 6-119                                  |
| DEACT   | = ***** GX | 6-121                                  |
| EXSIZ   | = ***** GX | 6-114                                  |
| HEADR   | = ***** GX | 5-84      7-175                        |
| MESSAGE | = 000000   | 7-157                                  |
| NBIAS   | = ***** GX | 7-161      7-164      8-212      8-217 |
| QBIAS   | = ***** GX | 7-172      7-179      8-209      8-219 |
| QSTRT   | = ***** GX | 8-210      8-215                       |
| R\$SMPL | = *****    | 5-71      6-110                        |
| R\$11M  | = 000000   | 5-68      6-106      7-157             |
| \$ALL16 | 000000 RG  | #5-67                                  |
| \$ALL18 | 000150 RG  | #7-168                                 |
| \$ALN18 | 000120 RG  | #7-157                                 |
| \$AMEM  | = ***** GX | 7-162      7-173                       |
| \$DEA16 | 000060 RG  | #6-105                                 |
| \$DEA18 | 000220 RG  | #8-205                                 |
| \$DECEX | 000064 RG  | #6-109                                 |
| \$DMEM  | = ***** GX | 8-214                                  |

CFGCONT CREATED BY MACRO ON 29-JUN-85 AT 00:01 PAGE 2 C 4

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                          |
|---------|------------|-------------------------------------|
| \$RAD50 | = 000016   | #8-123                              |
| \$RONLY | = *****    | 8-123 8-123                         |
| \$STRNG | = 000004   | #8-123                              |
| \$SUBXP | = 000010   | #8-123                              |
| \$CSR   | = ***** GX | *9-208                              |
| \$PRI   | = ***** GX | *9-225                              |
| \$VECT  | = ***** GX | *9-185                              |
| \$FLG   | = 177777   | #8-123                              |
| \$KEY   | = 177777   | #8-123                              |
| .PNUMB  | = ***** GX | 9-170 9-181 9-183 9-185 9-220 9-222 |
| .PNUMH  | = ***** GX | 9-179                               |
| .TPARS  | = ***** GX | 7-110                               |



\*\*\*FILE\*\*ID\*\*CFGDTE

C 5

```
CCCCCCCC  FFFFFFFFFF  GGGGGGGG  DDDDDDDD  TTTTTTTTTT  EEEEEEEEEE
CCCCCCCC  FFFFFFFFFF  GGGGGGGG  DDDDDDDD  TTTTTTTTTT  EEEEEEEEEE
CC         FF         GG         DD         TT         EE
CC         FF         GG         DD         TT         EE
CC         FF         GG         DD         TT         EE
CC         FF         GG         DD         TT         EE
CC         FF         GG         DD         TT         EE
CC         FFFFFFFF  GG         DD         TT         EEEEEEEE
CC         FFFFFFFF  GG         DD         TT         EEEEEEEE
CC         FF         GG  GGGGGG  DD         TT         EE
CC         FF         GG  GGGGGG  DD         TT         EE
CC         FF         GG   GG      DD         TT         EE
CC         FF         GG   GG      DD         TT         EE
CCCCCCCC  FF         GGGGGG  DDDDDDDD  TT         EEEEEEEEEE
CCCCCCCC  FF         GGGGGG  DDDDDDDD  TT         EEEEEEEEEE
```

```
LL         SSSSSSSS  TTTTTTTTTT
LL         SSSSSSSS  TTTTTTTTTT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LL         SSSSSS    TT
LL         SSSSSS    TT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT
```

CFGDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:02 Page 15  
DTE\$DF ACTION ROUTINES

```
341  
342          ; END OF DTE$DF MACRO  
343  
344 000342    DTEEND: CALL   CHKDTE          ; CHECK FOR VALID DTE ADDRESS  
345 000346    BCS      101$          ; ERROR - DTE ADDRESS NOT FOUND  
346 000350    RETURN  
347  
348          ; ERRORS  
349  
350 000352    101$:  MSG$R  03          ; ILLEGAL DTE ADDRESS
```

CFG  
Tab

CFGFEA - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 6 7  
Table of contents 00:03

5- 55 MACRO DEFINITIONS  
6- 73 \$QFEA - LOOK FOR FEASDF MACRO  
7- 102 FEASDF STATE TABLE  
8- 140 FEASDF ACTION ROUTINES

```
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113
```

```
.SBTTL LOCAL SYMBOL DEFINITIONS  
:  
: LOCAL SYMBOL DEFINITIONS  
:  
SPACE = 40 ; ASCII SPACE  
PVNAMX = 6 ; MAXIMUM LENGTH OF CIRCUIT ID  
  
:  
: SAVED REGISTER OFFSETS ON STACK FOR SWSTK$  
:  
R$R0 = 2 ; SAVED R0  
R$R1 = 4 ; SAVED R1  
R$R2 = 6 ; SAVED R2  
R$R3 = 10 ; SAVED R3  
R$R4 = 12 ; SAVED R4  
R$R5 = 14 ; SAVED R5
```

```
000040  
000006  
  
000002  
000004  
000006  
000010  
000012  
000014
```

```

586 .SBTTL INIXCB - INITIALIZE X25 CIRCUIT BLOCK
587
588
589
590 INIXCB - INITIALIZE X25 CIRCUIT BLOCK
591
592 INPUTS:
593     R0 - UNMAPPED ADDRESS OF CIRCUIT BLOCK
594
595 OUTPUTS:
596     R3 - MAPPED ADDRESS OF X25 CIRCUIT BLOCK
597     X25 CIRCUIT BLOCK INITIALIZED ACCORDING TO VALUES
598         SPECIFIED IN PVC$DF
599     R0,R1,R2 DESTROYED
600
601
602 INIXCB: MOV     R0,-(SP)      ;; GET UNMAPPED ADDRESS OF CIRCUIT BLOCK
603          CALL    $CEACX     ;; MAP TO CIRCUIT BLOCK
604          MOV     (SP)+,R3    ;; RETRIEVE MAPPED CIRCUIT BLOCK ADDRESS
605          MOV     #X$LEN,R1   ;; GET SIZE OF CIRCUIT BLOCK
606          MOV     R3,R2       ;; GET CIRCUIT BLOCK ADDRESS
607          CLR     (R2)+       ;; INITIALIZE CIRCUIT BLOCK TO ZEROES
608          DEC     R1           ;; MORE TO INITIALIZE?
609          BGT     10$         ;; BR IF YES
610          MOV     R3,R0       ;; COPY VIRTUAL ADDRESS OF XCB
611          CALL    BIASX      ;; BIAS CORRECTLY FOR APR6
612          MOV     PVCWSZ,X$WSZ(R3) ;; STORE WINDOW SIZE
613          MOV     PVCPSZ,X$PKSZ(R3) ;; STORE PACKET SIZE
614          MOV     PVCCTM,X$CTIM(R3) ;; STORE COUNTER TIMER
615          MOV     DTEDES,X$DTE(R3) ;; STORE DTE DESCRIPTOR ADDRESS
616          MOV     PVCPRR,X$PRR(R3) ;; STORE PORT NUMBER
617          MOV     PVCLCN,X$LCN(R3) ;; STORE LOGICAL CHANNEL NUMBER
618          MOV     R0,X$TXQ+2(R3) ;; INITIALISE TRANSMISSION QUEUE
619          ADD     #X$TXQ,X$TXQ+2(R3) ;;
620          MOV     R0,X$RXQ+2(R3) ;; INITIALISE RECEIVED DATA QUEUE
621          ADD     #X$RXQ,X$RXQ+2(R3) ;;
622          MOV     R0,X$WAQ+2(R3) ;; INITIALISE ACKNOWLEDGEMENT QUEUE
623          ADD     #X$WAQ,X$WAQ+2(R3) ;;
624          MOV     R0,X$ABQ+2(R3) ;; INITIALISE ABORTED DATA QUEUE
625          ADD     #X$ABQ,X$ABQ+2(R3) ;;
626          MOV     SP$P4,X$ST(R3) ;; SET PORT IN DATA TRANSFER STATE
627          MOV     #XT$PVC,X$TYP(R3) ;; SET CIRCUIT TYPE AS PVC
628          RETURN
  
```

CFGPVC      CREATED BY    MACRO    ON 29-JUN-85 AT 00:04      PAGE 4      C 10

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|---|
| SF.SER  | = 000001   | #5-70   |
| SF.SVC  | = 000002   | #5-70   |
| SF.UNL  | = 000040   | #5-70   |
| SLTMA   | = ***** GX | 21-775      21-779  |
| SPACE   | = 000040   | #6-102      11-303  |
| SP\$P4  | 000050 RG  | #7-154      17-626  |
| SYNERR  | = ***** GX | *9-192      *9-195      *20-745   |
| S\$BAS  | = *****    | 8-161      8-161      8-162      8-162      8-163      8-167      8-164      8-164      8-165 |
|         |            | 8-165      8-166      8-166      8-167      8-167      8-168      8-168                       |
| S.COST  | 000001     | #5-70   |
| S.FLG   | 000000     | #5-70   |
| S.LEN   | 000004     | #5-70   |
| S.NMST  | 000002     | #5-70   |
| S.OWNR  | 000003     | #5-70   |
| T\$FLTC | 000044     | #5-66      *21-817  |
| T\$LIF  | 000013     | #5-66   |
| T\$LIFL | 000013     | #5-66   |
| T\$LIFO | 000013     | #5-66   |
| T\$LIFS | 000013     | #5-66   |
| T\$LIN  | 000000     | #5-66   |
| T\$LIPS | 000006     | #5-66   |
| T\$LLD  | 000012     | #5-66   |
| T\$LLDC | 000045     | #5-66   |
| T\$LLDL | 000012     | #5-66   |
| T\$LLDO | 000012     | #5-66   |
| T\$LLDS | 000012     | #5-66   |
| T\$LEN  | 000046     | #5-66      21-812   |
| T\$LOPR | 000002     | #5-66   |
| T\$LTCL | 000024     | #5-66   |
| T\$LTIM | 000026     | #5-66   |
| T\$LTPR | 000014     | #5-66   |
| T\$LTPS | 000020     | #5-66   |
| T\$NAPL | 000004     | #5-66   |
| T\$NFE  | 000000     | #5-66   |
| T\$NLEN | 000010     | #5-66   |
| T\$NNUL | 000002     | #5-66   |
| T\$NOPL | 000006     | #5-66   |
| T\$NRNI | 000042     | #5-66   |
| T\$NRPL | 000005     | #5-66   |
| T\$NRUL | 000007     | #5-66   |
| T\$NVR  | 000001     | #5-66   |
| T\$RPR  | 000040     | #5-66   |
| T\$SVC  | 000034     | #5-66      *21-816  |
| T\$T5   | 000030     | #5-66      *21-818  |
| T\$T6   | 000032     | #5-66      *21-819  |
| XT\$PVC | = 000001   | 17-627  |
| X\$ABQ  | 000076     | *17-624      17-625      *17-625  |
| X\$CTIM | 000060     | *17-614   |
| X\$DTE  | 000024     | *17-615   |
| X\$LCN  | 000026     | *17-617   |
| X\$LEN  | 000106     | 14-481      17-605  |
| X\$MOWN | 000023     | *14-499   |

CFGSL  
Symbol

A\$SCH  
A\$SCF  
A\$SPR  
A\$STR  
BITS  
BITS1  
BITS2  
BRADJ  
BRADJ  
CFGBF  
CFGFL  
CFGSL  
CTL  
C\$CK  
C\$OF  
C\$RS  
D\$BL  
D\$IS  
D\$LL  
D\$YN  
D\$YN  
END  
E\$XF  
FG.X  
F\$LV  
G\$TH  
G\$IS  
G\$T  
G\$W  
I\$SR  
I\$SR  
K\$CH  
K\$CH  
K\$CH

. AB

\$STA  
\$TAF  
\$KST  
Error

\*\*\*

Work  
Work  
Size  
Oper

Elap  
SY:0

CFGSLT - SCAN CONFIGURATION FILE MACRO V05.03b Saturday 29-Jun-85 00:05 Page 9-1  
Symbol table

|                  |                    |                |                  |                     |
|------------------|--------------------|----------------|------------------|---------------------|
| ASSCHK= 000000   | K\$TPS= 000074     | L\$11R= 000000 | R\$DER= 000000   | T\$KMG= 000000      |
| ASSCPS= 000000   | LD\$LP= 000000     | L.COST 000015  | R\$K11= 000001   | T\$MIN= 000000      |
| ASSPRI= 000000   | LF.ACT= 100000     | L.CTL 000012   | R\$SND= 000000   | UNT = ***** GX      |
| ASSTRP= 000000   | LF.BRO= 000400     | L.CVA 177776   | R\$11M= 000000   | V\$SCTR= 001000     |
| BITS 000150R     | 002 LF.BWT= 000007 | L.DDM 000002   | SCHEL 000112R    | 002 X\$SDBT= 000000 |
| BIT1 000170R     | 002 LF.ENA= 002000 | L.DDS 000004   | SCHELO 000102R   | 002 \$ALPHA= 000022 |
| BIT2 000204R     | 002 LF.LPB= 001000 | L.DLC 000003   | SCTIM 000070R    | 002 \$ANY = 000020  |
| BRADJ 000122R    | 002 LF.MDC= 000100 | L.DLM 000006   | SF.ACT= 000200   | \$BLANK= 000006     |
| BRADJ1 000134R   | 002 LF.MFL= 004000 | L.DLS 000010   | SF.ENA= 000100   | \$CTIM = ***** GX   |
| CFGBF = ***** GX | LF.MTP= 000020     | L.FLG 000000   | SF.LPB= 000004   | \$DIGIT= 000024     |
| CFGFLG= ***** GX | LF.PAC= 000200     | L.KRBA 000016  | SF.MFL= 000040   | \$DNUMB= 000014     |
| CFGSZ = ***** GX | LF.RDY= 040000     | L.LEN = 000022 | SF.PAC= 000020   | \$EOS = 000012      |
| CTL = ***** GX   | LF.REA= 010000     | L.MPF 000022   | SF.REA= 000010   | \$ERRR= ***** GX    |
| C\$CKP= 000000   | LF.SER= 000040     | L.NMST 000020  | SF.SER= 000001   | \$ERRZP= ***** GX   |
| C\$ORE= 000400   | LF.TIM= 000010     | L.NSTA 000014  | SF.SVC= 000002   | \$ERR1T= ***** GX   |
| C\$RSH= 177564   | LF.UNL= 020000     | L.OWNR 000021  | SF.UNL= 000040   | \$EXIT = 000000     |
| D\$BUG= 177514   | LF.X2P= 000000     | L.UNT 000013   | SLIDF 000000R    | 002 \$FAIL = 177777 |
| D\$ISK= 000000   | LN.CLO= 000000     | M\$CRB= 000124 | SLTKW 000000RG   | 003 \$LAMP= 000000  |
| D\$L11= 000001   | LN.DUM= 000005     | M\$CRX= 000000 | SLTST 000000RG   | 002 \$NUMB= 000002  |
| D\$YNC= 000000   | LN.LOA= 000004     | M\$FCS= 000000 | SYNERR= ***** GX | \$QSLT 000000RG     |
| D\$YNM= 000000   | LN.LOD= 000003     | M\$MGE= 000000 | S\$WRG= 000000   | \$RAD50= 000016     |
| END 000220R      | 002 LN.OAU= 000003 | M\$NET= 000000 | S\$YSZ= 007600   | \$STRNG= 000004     |
| E\$XPR= 000000   | LN.OFF= 000001     | M\$OVR= 000000 | S.BRAD 000210R   | \$SUBXP= 000010     |
| FG.X2P= ***** GX | LN.ON = 000000     | M\$ACC= 000001 | S.COST 000001    | \$HTIM= ***** GX    |
| F\$LVL= 000001   | LN.OOP= 000004     | N\$BUB= 000001 | S.CTIM 000126R   | \$LTIM= ***** GX    |
| G\$TPP= 000000   | LN.OPE= 000001     | N\$LDV= 000001 | S.CTL 000062R    | \$SFLG= 177777      |
| G\$ISS= 000000   | LN.REF= 000002     | N\$MCP= 000001 | S.FLG 000000     | \$SKEY= 000002      |
| G\$TTK= 000000   | LN.SER= 000002     | N\$MLL= 000001 | S.HTIM 000152R   | \$SSTA= 000000      |
| G\$WRD= 000000   | LN.STA= 000017     | N\$MOV= 000010 | S.LEN 000004     | \$STMP= 000016R     |
| I\$RAR= 000000   | LN.SUB= 000360     | N\$NCT= 000001 | S.NAME= ***** GX | .NBRA = ***** GX    |
| I\$RDN= 000000   | LN.TRI= 000006     | N\$PEM= 000001 | S.NMST 000002    | .PNUMB= ***** GX    |
| K\$CNT= 177546   | L\$ASG= 000000     | P\$P45= 000000 | S.OWNR 000003    | .PNUMH= ***** GX    |
| K\$CSR= 177546   | L\$DPV= 000000     | P\$WRD= 000000 | S.RPRI 000220R   | .RPRI = ***** GX    |
| K\$LDC= 000000   | L\$P11= 000001     | Q\$OPT= 000010 | S.UNT 000104R    | .TPARS= ***** GX    |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
000252 001 (RW,I,LCL,REL,CON)  
\$STATE 000230 002 (RW,D,LCL,REL,CON)  
\$KTAB 000006 003 (RW,D,LCL,REL,CON)  
\$KSTR 000025 004 (RW,D,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistic

Work file reads: 0  
Work file writes: 0  
Size of work file: 13185 Words ( 52 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: R3X-11M/PLUS

Elapsed time: 00:00:46.21

SY:CFGSLT.V2,[132,134]CFGSLT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFGSLT

CFGSTA - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 00:06 Page 10  
STA\$DF ACTION ROUTINES

.SBTTL STA\$DF ACTION ROUTINES

```

175
176
177
178      ; STATION ADDRESS (STA$DF)
179
180 000270 005767 000000G ST.NUM: TST .PNUMH ; MAKE SURE STATION ADDRESS
181 000274 001022      BNE 101$ ; IS LESS THAN 256.
182 000276 105:67 000001G      TSTB .PNUMB+1 ; ...
183 000302 001017      BNE 101$ ;
184 000304 016700 000000G      MOV SCNT,R0 ; GET CURRENT STATION COUNT
185 000310 022700 000060      CMP #48.,R0 ; HAVE WE ALREADY SEEN ALL 48.?
186 000314 101415      BLOS 111$ ; IF LOS, YES
187 000316 1:6760 000000G 000000G      MOVB .PNUMB,$$SNUM(R0) ; STORE STATION ADDRESS IN TABLE
188 000324 0C1406      BEQ 101$ ; IF ZERO, ILLEGAL STATION ADDRESS
189 000326 005267 000000G      INC SCNT ; INCREMENT STATION COUNT
190 000332 016767 000000G 000000G      MOV SCNT,$$MTP ; SAVE IT FOR TEMPLATE ROUTINES
191 000340      RETURN
192 000342      101$: EMSG$R 1A ; ILLEGAL STATION ADDRESS
193 000350      111$: EMSG$R 1B ; MORE THAN 48. STA$DF'S
194
195      ; STATION COST (STA$DF)
196
197
198 000356 026727 000000G 000031 ST.CST: CMP .PNUMB,#MAXCST ; IS THE COST IN RANGE ?
199 000364 101025      BHI 101$ ; IF HI, NO .. ERROR
200 000366 016700 000000G      MOV $SLTA,R0 ; GET THE SLT ADDRESS
201 000372 032760 000020 000000      BIT #LF.MTP,L.FLG(R0) ; IS THIS A MULTIPOINT LINE?
202 000400 001416      BEQ 10$ ; BR IF NO
203 000402 105760 000014      TSTB L.NSTA(R0) ; IS THIS A MASTER LINE ?
204 000406 001413      BEQ 10$ ; IF EQ, NO .. TREAT AS PT-TO-PT LINE
205 000410 016701 000000G      MOV SCNT,R1 ; GET THE STATION COUNT
206 000414 005301      DEC R1 ; REALLY LAST ONE
207 000416 006301      ASL R1 ; CONVERT LOGICAL STATION NUMBER TO 2 WORD INDEX
208 000420 006301      ASL R1
209 000422 060001      ADD R0,R1 ; GET START OF STATION TABLE
210 000424 062701 000022      ADD #L.MPF,R1
211 000430 116761 000000G 000001      MOVB .PNUMB,S.COST(R1) ; STORE STATION COST
212 000436      10$: RETURN
213
214 000440      101$: EMSG$R YP ; ILLEGAL LINE COST
215
216      ; STATION ACTIVE POLLING RATIO
217
218
219 000446      ST.APR:
220 000446 105767 000001G      TSTB .PNUMB+1 ; IS IT A BYTE VALUE ?
221 000452 001007      BNE 101$ ; IF NE, NO .. ERROR
222 000454 016700 000000G      MOV SCNT,R0 ; GET CURRENT STATION COUNT
223 000460 005300      DEC R0 ; REALLY LAST ONE
224 000462 116760 000000G 000000G      MOVB .PNUMB,$$APR(R0) ; SAVE ACTIVE POLLING RATIO
225 000470      RETURN
226
227 000472      101$: EMSG$R YQ ; ILLEGAL ACTIVE POLLING RATIO

```



CFG SVC - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 00:06 Page 8  
 \$Q SVC - LOOK FOR SVC\$DF MACRO

```

152 .SBTTL $Q SVC - LOOK FOR SVC$DF MACRO
153
154 ;+
155 ; $Q SVC - LOOK FOR SVC$DF MACRO
156
157 INPUTS:
158 NONE
159
160 OUTPUTS:
161 ALL REGISTERS DESTROYED
162
163 -
164
165 000000 012705 000000' $Q SVC:: MOV #SVCDF,R5 ; STATE TABLE ADDRESS
166 000004 005001 CLR R1 ; FULL KEYWORD MATCH LENGTH
167 000006 012702 000000' MOV #SVCKW,R2 ; KEYWORD TABLE ADDRESS
168 000012 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
169 000016 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
170 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
171 000026 CALL TPARS ; GO DO THE PARSE
172 000032 103003 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
173 000034 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
174 000040 001401 BEQ 101$ ; IF EQ, YES
175 000042 20$: RETURN
176
177 ; ERRORS
178
179 000044 101$: EMSG$R 1T ; SYNTAX ERROR

```

|                  |                     |                 |                  |                     |
|------------------|---------------------|-----------------|------------------|---------------------|
| A\$CHK= 000000   | H\$CUG 000010       | LN.OFF= 000001  | N\$RT2 000006    | PT\$LV2= 000001     |
| A\$CPS= 000000   | H\$DST 000012       | LN.ON = 000000  | N\$RTCL 000112   | PT\$PH3= 000010     |
| A\$PRI= 000000   | H\$D29 000014       | LN.OOP= 000004  | N\$TLCL 000100   | PT\$XAR= 000020     |
| A\$TRP= 000000   | H\$FLG 000000       | LN.OPE= 000001  | N\$TNC 000114    | PX\$BLK= 000040     |
| CFERR = ***** GX | H\$GLEN 000104      | LN.REF= 000002  | N\$TRC 000106    | PX\$DLM= 000200     |
| CFGBF = ***** GX | H\$GLT 000044       | LN.SER= 000002  | N\$TTCB 000110   | PX\$SVC= 000100     |
| CFGSZ = ***** GX | H\$GNAM 000050      | LN.STA= 000017  | N\$VER 000066    | P\$CHN 000004       |
| COUNT 000024R    | 002 H\$GNML= 000020 | LN.SUB= 000360  | N\$XLEN 000124   | P\$CNT 000005       |
| CTL\$VC 000021R  | 002 H\$GPT 000046   | LN.TRI= 000006  | N\$XACC= 000001  | P\$CTR 000034       |
| CVTBUF 000054R   | 002 H\$HITS 000034  | LTIMR 000004R   | N\$XBUF= 000001  | P\$DRCT 000015      |
| C\$CKP= 000000   | H\$HLEN 000044      | L\$ASG= 000000  | N\$XLDV= 000001  | P\$DRTR 000036      |
| C\$SORE= 000400  | H\$LBDA 000070      | L\$DRV= 000000  | N\$X MCP= 000001 | P\$FLG 000012       |
| C\$RSH= 177564   | H\$LBDN 000072      | L\$P11= 000001  | N\$XMLL= 000001  | P\$FWD 000030       |
| DTEEND= 000044R  | 002 H\$LDTE 000002  | L\$11R= 000000  | N\$XMOV= 000010  | P\$ICCB 000046      |
| DTELEN= 000017   | H\$LEN 000042       | L.COST 000015   | N\$XNCT= 000001  | P\$IPL 000014       |
| DTEPCX 000044R   | 002 H\$LDTS 000032  | L.CTL 000012    | N\$XPEM= 000001  | P\$LCD 000002       |
| DTE\$VC 000025R  | 002 H\$NETW 000024  | L.CVA 177776    | OWN\$VC 000020R  | P\$LEN 000052       |
| D\$BUG= 177514   | H\$NML = 000006     | L.DDM 000002    | PCKBCD 001440R   | P\$LST 000000       |
| D\$ISK= 000000   | H\$NPT 000022       | L.DDS 000004    | PDVID = ***** GX | P\$NRNI 000040      |
| D\$LL1= 000001   | H\$PTB 000020       | L.DLC 000003    | PDVTA = ***** GX | P\$OCB 000050       |
| D\$SYNC= 000000  | H\$PVC 000006       | L.DLM 000006    | PF\$BLK= 020000  | P\$PFQ 000006       |
| D\$SYNM= 000000  | H\$RDTE 000004      | L.DLS 000010    | PF\$CLC= 010000  | P\$PKSZ 000044      |
| E\$NBR 000014    | H\$RNW 000042       | L.FLG 000000    | PF\$DLM= 100000  | P\$RMX1 000016      |
| E\$NBS 000020    | H\$SVC 000036       | L.KRBA 000016   | PF\$EIP= 000002  | P\$RMX2 000020      |
| E\$NCR 000034    | H\$TRB 000016       | L.LEN = 000022  | PF\$ENB= 000001  | P\$RPR1 000042      |
| E\$NCS 000036    | H\$XAVL 000100      | L.MPF 000022    | PF\$FAI= 004000  | P\$RTIM 000003      |
| E\$NIC 000044    | H\$XBIA 000074      | L.NMST 000020   | PF\$FM1= 000100  | P\$STA1 000022      |
| E\$NLEN 000050   | H\$X29C 000040      | L.NSTA 000014   | PF\$FM2= 000200  | P\$STA2 000023      |
| E\$NLLA 000012   | I\$XRAR= 000000     | L.OWNR 000021   | PF\$OFF= 000000  | P\$TIM 000010       |
| E\$NLNK 000000   | I\$SRDN= 000000     | L.UNT 000013    | PF\$RM1= 000020  | P\$TSC1 000026      |
| E\$NML 000040    | KSAR5 = ***** GX    | MAXWND= 000007  | PF\$RM2= 000040  | P\$TSIZ 000024      |
| E\$NMR 000024    | K\$CNT= 177546      | M\$SCRB= 000124 | PF\$RVR= 000010  | P\$TYP 000001       |
| E\$NMS 000030    | K\$CSR= 177546      | M\$SCRX= 000000 | PF\$STA= 000004  | P\$P45= 000000      |
| E\$NNOD 000002   | K\$LDLC= 000000     | M\$SCFS= 000000 | PF\$SVC= 040000  | P\$WRD= 000000      |
| E\$NRT 000042    | K\$STPS= 000074     | M\$SMGE= 000000 | PF\$UP = 000004  | Q\$SPT= 000010      |
| E\$NRTP 000005   | LD\$LP = 000000     | M\$NET= 000000  | PKTSVC 000006R   | 002 RCL\$VC 000023R |
| E\$NSEG 000010   | LF.ACT= 100000      | M\$OVR= 000000  | PR\$BED= 000200  | RF.CTL= 000003      |
| E\$NTIM 000046   | LF.BRO= 000400      | NEXT 000000R    | PR\$BEU= 000100  | RF.LD1= 000040      |
| E\$NUSE 000004   | LF.BWT= 000007      | NODTE 000420R   | PR\$BRD= 000040  | RF.LD2= 000100      |
| E\$STRT 000006   | LF.ENA= 002000      | N\$ADJ1 000072  | PR\$BRU= 000020  | RF.TIM= 177400      |
| E\$XPR= 000000   | LF.LPB= 001000      | N\$ADJ2 000074  | PR\$DWN= 000002  | RF.TMO= 000400      |
| FLGSVC 000010R   | 002 LF.MDC= 000100  | N\$CACH 000062  | PR\$LCC= 000010  | RF.WFC= 000200      |
| FMTR = ***** GX  | LF.MFL= 004000      | N\$CRC 000120   | PR\$MOP= 000004  | RF.WTD= 000020      |
| FMTRB = ***** GX | LF.MTP= 000020      | N\$HC1 000052   | PR\$UP = 000001  | RF.WTM= 000030      |
| FM.8 = 000000    | LF.PAC= 00020C      | N\$HC2 000056   | PSIPT = ***** GX | RF.WTS= 000010      |
| FM.8B = 000000   | LF.RDY= 040000      | N\$LV1 000002   | PS\$CHR= 000016  | RTSPC = ***** GX    |
| FNDOWN 001146R   | LF.REA= 010000      | N\$LV2 000010   | PS\$FAI= 000014  | RTY\$VC 000012R     |
| F\$LV1= 000001   | LF.SER= 000040      | N\$MHC1 000036  | PS\$NTI= 000006  | R\$D\$R= 000000     |
| G\$STP= 000000   | LF.TIM= 000010      | N\$MHC2 000044  | PS\$OFF= 000000  | R\$K11= 000001      |
| G\$STSS= 00000C  | LF.UNL= 020000      | N\$PLD 000016   | PS\$STR= 000002  | R\$SND= 000000      |
| G\$STTK= 000000  | LF.X2P= 000000      | N\$PRI 000076   | PS\$UP = 000012  | R\$11M= 000000      |
| G\$WRD= 000000   | LLCTA = ***** GX    | N\$ROA1 000022  | PS\$VER= 000010  | SETTLC 001202R      |
| HF\$DLM= 000002  | LN.CLO= 000000      | N\$ROA2 000030  | PS\$WT = 000004  | SF\$BLK= 000001     |
| HF\$GWY= 000010  | LN.DUM= 000005      | N\$RTMX 000014  | PT\$BRO= 000200  | SF\$INC= 000002     |
| HF\$HOS= 000004  | LN.LCA= 000004      | N\$RTM1 000014  | PT\$DRT= 000100  | SF.ACT= 000200      |
| HF\$XDF= 000020  | LN.LDO= 000003      | N\$RTM2 000015  | PT\$END= 000004  | SF.ENA= 000100      |
| HTIMR 000002R    | 002 LN.DAU= 000003  | N\$RT1 000000   | PT\$LV1= 000002  | SF.LPB= 000001      |

```
101 .SBTTL ERROR MESSAGES
102
103 ;
104 ; ERROR MESSAGES
105 ;
106
107 000000 .PSECT DATA,D
108
109 .ENABL LC
110 000000 NTLERS ,1G,8B,CERR,RTSPC,CFLIN,<Unit Number>
111 000024 NTLERS ,1H,8,CERR,RTSPC,CFLIN,<UNIT$DF Out Of Order>
112 000060 NTLERS ,1K,8B,CERR,RTSPC,CFLIN,<Characteristics Value>
113 000116 NTLERS ,1L,8B,CERR,RTSPC,CFLIN,<Secondary CSR Address>
114 000154 NTLERS ,1M,8,CERR,RTSPC,CFLIN,<Secondary CSR Not Allowed>
115 000216 NTLERS ,1N,8,CERR,RTSPC,CFLIN,<Secondary CSR Missing>
116 000254 NTLERS ,ZO,8A,CERR,REP8A,CFLIN,<Secondary CSR Offline>
117 .DSABL LC
118
119 000040 .PSECT
```

CFGUNT CREATED BY MACRO ON 15-JUL-85 AT 18:57 PAGE 2 C 16

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE    | REFERENCES  |
|--------|----------|---|
| L.CVA  | 177776   | #5-60   |
| L.DDM  | 000002   | #5-60 12-303  |
| L.DDS  | 000004   | #5-60   |
| L.DLC  | 000003   | #5-60 12-303  |
| L.DLM  | 000006   | #5-60   |
| L.DLS  | 010010   | #5-60   |
| L.FLG  | 010000   | #5-60 13-392  |
| L.KRBA | 000016   | #5-60   |
| L.LEN  | = 000022 | #5-60   |
| L.MPF  | 000022   | #5-60 13-394  |
| L.NMST | 000020   | #5-60   |
| L.NSTA | 000014   | #5-60   |
| L.OWNR | 000021   | #5-60   |
| L.UNT  | 000013   | #5-60 11-233 11-256 11-271 12-289 13-356 13-388                     |
| MAXCST | = 000031 | #6-92 13-383  |
| MS.SEC | = *****  | GX 13-352   |
| RDBSZ  | = *****  | GX 12-311   |
| REP8A  | = *****  | GX 7-116  |
| RTSPC  | = *****  | GX 7-110 7-111 7-112 7-113 7-114 7-115                              |
| SETTMO | = 000400 | R 11-254 #12-284  |
| SF.ACT | = 000200 | #5-60   |
| SF.ENA | = 000100 | #5-60   |
| SF.LPB | = 000004 | #5-60   |
| SF.MFL | = 000040 | #5-60   |
| SF.PAC | = 000020 | #5-60   |
| SF.REA | = 000010 | #5-60   |
| SF.SER | = 000001 | #5-60   |
| SF.SVC | = 000002 | #5-60   |
| SF.UNL | = 000040 | #5-60   |
| SPEEDT | = 000000 | R #6-73 12-300  |
| SYNERR | = *****  | GX *8-143   |
| SSBAS  | = *****  | GX 7-110 7-111 7-112 7-113 7-114 7-115 7-116                        |
| S.COST | 000001   | #5-60 *13-395   |
| S.FLG  | 000000   | #5-60   |
| S.LEN  | 000004   | #5-60   |
| S.NMST | 000002   | #5-60   |
| S.OWNR | 000003   | #5-60   |
| UCNT   | = *****  | GX 11-219 11-230 11-253 11-269 13-341 13-356 *13-361 13-374 *13-376 |
| UNTDI  | 000000   | R 13-386  |
| UNTKW  | 000000   | RG 8-135  |
| UNTST  | 000000   | RG #9-153   |
| U.CHA0 | 000156   | R #11-228   |
| U.CHA1 | 000262   | R #11-251   |
| U.CST  | 001022   | R #13-383   |
| U.DPR  | 001114   | R #13-405   |
| U.PECH | 000334   | R #11-267   |
| U.SLSR | 000614   | R #13-339   |
| U.UNT  | 000112   | R #11-215   |
| U.XCSR | 000756   | R #13-370   |
| SAIPHA | = 000022 | #9-153  |

EVL            CREATED BY MACRO ON 28-JUN-85 AT 22:58            PAGE 5            D 1  
 SYMBOL CROSS REFERENCE            CREF    04.00

| SYMBOL   | VALUE      | REFERENCES                                      |
|----------|------------|---|
| NETIME   | 000426 R   | 7-81 #10-291                                    |
| NS\$VCT  | = *****    | 9-174 13-421 13-462 13-473                      |
| OP\$INI  | = 000000   | #5-61   |
| OP\$TER  | = 000001   | #5-61   |
| OS\$ACHD | 000012     | 13-464  |
| PDVID    | 000030 R   | #8-108 10-295                                   |
| PDVTA    | 000032 R   | #8-109 10-299 12-398                            |
| PH\$HDE  | = 000004   | #5-61   |
| PH\$LOC  | = 000002   | #5-61   |
| PH\$MTS  | = 000003   | #5-61   |
| PH\$UMP  | = 000000   | #5-61   |
| PH\$WCS  | = 000001   | #5-61   |
| PR7      | = ***** GX | 9-199 11-338                                    |
| PS       | = ***** GX | 9-199 *9-199 *9-223 11-338 *11-338 *11-368      |
| PSIPT    | 000034 R   | #8-110 13-424 13-443                            |
| QUEUE    | 001132 R   | 9-211 11-353 #14-491                            |
| Q\$PORT  | 000003     | 13-468  |
| C\$STN   | 000002     | 13-470  |
| RE\$ADC  | = 000004   | #5-61   |
| RE\$ADF  | = 000017   | #5-61   |
| RE\$ADR  | = 000007   | #5-61   |
| RE\$BLK  | = 000010   | #5-61   |
| RE\$CAF  | = 000014   | #5-61   |
| RE\$DAT  | = 000001   | #5-61   |
| RE\$DRP  | = 000016   | #5-61   |
| RE\$LDI  | = 000013   | #5-61   |
| RE\$LSN  | = 000012   | #5-61   |
| RE\$NML  | = 000001   | #5-61   |
| RE\$OPE  | = 000004   | #5-61   |
| RE\$OPR  | = 000000   | #5-61   |
| RE\$RCV  | = 000001   | #5-61   |
| RE\$SED  | = 000011   | #5-61   |
| RE\$SKW  | = 000006   | #5-61   |
| RE\$STA  | = 000002   | #5-61   |
| RE\$SUM  | = 000003   | #5-61   |
| RE\$SYN  | = 000000   | #5-61   |
| RE\$TME  | = 000021   | #5-61   |
| RE\$TMO  | = 000000   | #5-61   |
| RE\$TMR  | = 000020   | #5-61   |
| RE\$UPT  | = 000002   | #5-61   |
| RE\$URE  | = 000003   | #5-61   |
| RE\$VER  | = 000005   | #5-61   |
| RE\$VRO  | = 000015   | #5-61   |
| RT\$INI  | = 000002   | #5-61   |
| RT\$OFF  | = 000001   | #5-61   |
| RT\$ON   | = 000000   | #5-61   |
| R\$MPL   | = *****    | 8-86 8-89 8-102 9-201 9-213 9-226 11-340 11-358 |
| R\$M1D   | = *****    | 5-62  |
| R\$M1M   | = 000000   | 5-62  |
| R\$M1S   | = *****    | 5-62  |
| SC\$OFF  | = 000001   | #5-61   |
| SC\$ON   | = 000000   | #5-61   |

```

286 000264          40$:
287 000264 011200      MOV      (R2),R0
288 000266 001404      BEQ      50$
289
290
291
292
293
294
295
296 000270 020312      CMP      R3,(R2)
297 000272 103402      BLO      50$
298
299 000274 010015      MOV      R0,(R5)
300 000276 000772      BR       40$
301 000300
302
303
304
305
306
307
308 000300 005722      TST      (R2)+
309 000302 011546      MOV      (R5),-(SP)
310 000304 061216      ADD      (R2),(SP)
311 000306 022603      CMP      (SP)+,R3
312 000310 001002      BNE      60$
313 000312 061612      ADD      (SP),(R2)
314
315
316
317
318
319 000314 000404      BR       70$
320 000316
321
322
323
324
325
326 000316 005742      TST      -(R2)
327
328 000320 010312      MOV      R3,(R2)
329
330 000322 011215      MOV      (R2),(R5)
331 000324 010022      MOV      R0,(R2)+
332
333
334
335 000326          70$:
336
337
338
339
340
341
342 000326 011546      MOV      (R5),-(SP)
    
```

```

: SAVE NEXT HOLE POINTER.
: IS THIS THE LAST HOLE?
: THIS BRANCH IS TAKEN IF
: THE PARTITION WAS FULL
: INITIALLY [IN WHICH CASE
: (R5)=(KP.AR3)=R3= 1/64 TH THE
: BASE ADDRESS OF THE TASK SEGMENT;
: OR IF (R3) IS G.T. THE
: LAST HOLE.
: NO -- IS THE NEXT HOLE G.T. THE
: TASK SEGMENT HOLE-- I.E. IS IT THE
: MINIMUM UPPER BOUND?
: NO -- SET ASR TO NEXT HOLE
: AND CONTINUE SCAN.
: YES -- CHECK TO SEE IF THE LOWER
: BOUND [THE BASE ADDRESS OF WHICH
: IS (KP.AR3)] IS ADJACENT TO
: TASK SEGMENT HOLE. NOTE THAT IF THERE
: IS NO LOWER BOUND THE TASK SEGMENT
: HOLE WILL BE COMPARED TO ITSELF
: AND THE CHECK WILL FAIL.
: SET PTR TO LOWER HOLE SIZE.
: =BASE ADDRESS TO STACK
: = BASE + SIZE.
:
: ADJANCY?
: YES -- SET LOW HOLE SIZE = OLD
: SIZE + MAX TASK SEGMENT SIZE. THE
: TASK SEGMENT HOLE IS NOW GONE, HAVING
: BEEN ADJOINED TO ITS LOWER
: BOUND.
:
: GO TEST FOR HIGH ADJANCY
: NO -- LINK LOWER BOUND TO
: TASK HOLE. IF THE LOW BOUND
: IS TASK HOLE THIS CODE IS
: AN EFFECTIVE NOP AND THE 1ST
: GUESS CONDITIONS ABOVE REMAIN
: (AND OF COURSE ARE VALID).
: BACK UP TO NEXT HOLE POINTER.
: NOTE ALSO THAT (R2) = R5.
: THIS SETS LOW BOUND HOLE TO
: POINT TO TASK SEGMENT HOLE.
: SET ASR TO ACCESS TASK SEGMENT HOLE.
: THIS LINKS THE TASK SEGMENT HOLE TO
: THE HOLE THAT THE LOW BOUND
: USED TO POINT TO. NOTE THAT
: R2 POINTS TO SIZE I.E. = 60002.
: CHECK TO SEE IF LOWER BOUND
: (BASE ADDRESS IN KP.AR3) IS ADJACENT
: TO UPPER BOUND. IT SHOULD BE
: CLEAR THAT THE NON EXISTENT
: UPPER BOUND CASE (I.E. THE
: LOWER BOUND HOLE POINTER = 0)
: WILL FAIL THE ADJANCY TEST.
: =(KP.AR3)=LOW BASE ADDR TO STK
    
```

CEAL  
MACR  
MACR  
CALL  
RETU  
SWST

\*\*F

CEALL      CREATED BY MACRO ON 29-JUN-85 AT 00:01      PAGE 2      D 3

MACRO CROSS REFERENCE      CREF 04.00

MACRO NAME      REFERENCES

|         |       |       |       |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CALL    | 5-69  | 5-77  | 5-82  | 6-107 | 6-121 | 7-158 | 7-162 | 7-169 | 7-173 | 8-206 |
|         | 8-214 |       |       |       |       |       |       |       |       |       |
| RETURN  | 5-89  | 6-122 | 7-181 | 8-220 |       |       |       |       |       |       |
| SWSTK\$ | 5-69  | 6-107 | 7-158 | 7-169 | 8-206 |       |       |       |       |       |

CFG

MACR

MACR

ASL

CALL

DBG

EMSC

ISTA

MTR

NTLE

REJ

RETU

STA

TRA

CFGCNT      CREATED BY    MACRO    ON 29-JUN-85 AT 00:01      PAGE 3      D 4  
 MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES   |
|------------|--|
| ASL\$      | #5-57      9-224   |
| CALL       | 7-110      9-196      9-206  |
| DBGTF\$    | #8-123   |
| EMSG\$P    | #5-57      7-119      9-188      9-189      9-211      9-212      9-228  |
| ISTAT\$    | #5-57      8-123   |
| MTRAN\$    | #8-123   |
| NTLR\$     | #5-57      6-77      6-78      6-79      6-80      6-81  |
| REJ\$      | #5-65      9-172   |
| RETURN     | 7-114      9-173      9-187      9-210      9-227  |
| STAT\$     | #5-57      8-127      #8-129      #8-131      #8-133      #8-135      #8-137      #8-139      8-144              |
| TRANS      | #5-57      #8-128      #8-130      #8-132      #8-134      #8-136      #8-138      #8-140      #8-145      8-164 |



CFGDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:02 <sup>D E</sup>  
Table of contents

|     |     |  |
|-----|-----|--|
| 5-  | 54  | MACRO DEFINITIONS                            |
| 6-  | 78  | LOCAL SYMBOL DEFINITIONS                     |
| 7-  | 98  | LOCAL DATA                                   |
| 8-  | 125 | ERROR MESSAGES                               |
| 9-  | 140 | LOOK FOR DTE\$DF MACRO                       |
| 10- | 169 | TPARS STATE TABLES                           |
| 11- | 221 | LINE-ID SUBEXPRESSION                        |
| 12- | 244 | DTE\$DF ACTION ROUTINES                      |
| 16- | 352 | CHKDTE - CHECK FOR VALID DTE ADDRESS         |
| 17- | 395 | BIASX - SET APR6 BIAS CORRECTLY              |
| 18- | 416 | LINE-ID ACTION ROUTINES                      |
| 19- | 447 | STRNXT - STORE NEXT DIGIT                    |
| 20- | 473 | PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT |

.SBTTL CHKDTE - CHECK FOR VALID DTE ADDRESS

```

: +
: CHKDTE - CHECK FOR VALID DTE ADDRESS
: INPUTS:
:   NONE
: OUTPUTS:
:   CARRY CLEAR - DTE ADDRESS IS UNIQUE
:   R4 = COUNT OF DTEs DEFINED ALREADY
:   CARRY SET - DTE ADDRESS IS NOT UNIQUE
:   R0,R1,R2 DESTROYED
: -
CHKDTE: MOV    @PSIPT,R0      ; POINT TO PSI HOME BLOCK
        ADD    #H$LDTE,R0   ; POINT TO LOCAL DTE LISTHEAD

        SWSTK$ 40$          ; ENTER SYSTEM STATE
        SAVMAP          ; SAVE CURRENT MAPPING
        CLR      R4         ; INITIALISE DTE COUNT
10$:    MOV      (R0),R0     ; POINT TO NEXT BLOCK IN LIST
        BEQ      25$        ; BR IF END OF LIST
        MOV      R0,DTEDES  ; SAVE UNMAPPED ADDRESS OF DTE
        INC      R4         ; ONE MORE BLOCK SEEN
        MOV      R0, -(SP)  ; SET UP UNMAPPED ADDRESS
        CALL     $CEACX     ; CONVERT TO MAPPED ADDRESS
        MOV      (SP)+,R0   ; RETRIEVE MAPPED ADDRESS
        MOV      #<DTEALN+1>/2,R1 ; GET DTE ADDRESS LENGTH
        MOV      R0,R2     ; GET ADDRESS OF LOCAL DTE DESCRIPTOR
        ADD      #L$DTEA,R2 ; POINT TO DTE ADDRESS
        MOV      #DTEPCK,R3 ; POINT TO SPECIFIED DTE ADDRESS
20$:    CMPB     (R2)+,(R3)+ ; DO THE ADDRESSES MATCH?
        BNE      10$        ; BR IF NO
        DEC      R1         ; MORE OF ADDRESS TO CHECK?
        BGT      20$        ; BR IF YES
        BR       30$        ; BR IF FOUND IT
25$:    RETC      R0         ; INDICATE NOT FOUND
30$:    RESMAP          ; RESTORE PREVIOUS MAPPING
        MOV      R4,R$R4(SP) ; RETURN COUNT IN TASK R4
40$:    RETURN

```

```

352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368 000360 017700 000000G
369 000364 062700 000002
370
371 000370
372 000374
373 000400 005004
374 000402 011000
375 000404 001423
376 000406 010067 000000G
377 000412 005204
378 000414 010046
379 000416
380 000422 012600
381 000424 012701 000010
382 000430 010002
383 000432 062700 000020
384 000436 012703 000017'
385 000442 122223
386 000444 001356
387 000446 005301
388 000450 003374
389 000452 000405
390 000454
391 000466
392 000472 010466 000012
393 000476

```

.TITLE CFGFEA - CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 04-JUN-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/s v4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

115          .SBTTL LOCAL DATA
116
117          ;
118          ; LOCAL DATA
119          ;
120
121          .PSECT DATA,D
122
123          .NLIST BEX
124
125          ;
126          ; LOCAL DATA FOR PVC$DF MACRO
127          ;
128          000000 HTIMR: .BLKW 1 ; LOCAL COPY OF HELLO TIMER
129          000002 LTIMR: .BLKW 1 ; LOCAL COPY OF LISTEN TIMER
130          000004 PORTNO: .BLKW 1 ; NUMBER OF PORTS
131          000006 PRTADD: .BLKW 1 ; ADDRESS OF ENTRY IN PORT TABLE
132          000010 PVCLCN: .BLKW 1 ; LOGICAL CHANNEL NUMBER
133          000012 PVCCTM: .BLKW 1 ; COUNTER TIMER
134          000014 PVCPSZ: .BLKW 1 ; MAXIMUM BLOCK SIZE
135          000016 PVCWSZ: .BLKW 1 ; WINDOW SIZE
136          000020 PVCOWN: .BLKW 1 ; OWNER PROCESS
137          000022 DLMPDV: .BLKW 1 ; ADDRESS OF DLM PDV
138          000024 DLMSLT: .BLKW 1 ; ADDRESS OF DLM SLT
139          000026 DLMSTA: .BLKW 1 ; DLM'S STATION NUMBER
140          000030 DLMSLN: .BLKW 1 ; DLM'S SYSTEM LINE NUMBER
141          000032 FLAG: .BLKW 1 ; LOCAL FLAGS WORD
142          000001 000001 PF.DLM = 1 ; PVC IS A DLM CIRCUIT
143          000034 PVCPR1: .BLKB 1 ; PORT NUMBER OF FREE ENTRY
144          000035 PVCFLG: .BLKB 1 ; PVC FLAGS BYTE
145          000036 PVCNAM: .BLKB 6 ; CIRCUIT IDENTIFICATION
146          000044 DLMCTL: .BLKB 1 ; DLM'S CONTROLLER NUMBER
147          000045 000 DLMUNT: .BYTE 0 ; DLM'S UNIT NUMBER (ALWAYS 0, MUST BE MTP)
148          000046 DLMIND: .BLKB 1 ; DLM'S PDV INDEX
149          .EVEN
150
151          ;
152          ; STATE VALUE FOR "DATA TRANSFER STATE" FOR XCB
153          ; GLOBAL PATCHABLE JUST IN CASE ...
154          000050 000003 SP$P4: .WORD 3 ; VALUE FOR DATA TRANSFER STATE
155

```

```

630 .SBTTL FNDHSH - FIND HASH TABLE ENTRY
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646 FNDHSH: SAVMAP
647 001452 MOV DTEDES,-(SP)
648 001456 016746 $CEACX
649 001466 012602 CALL (SP)+,R2
650 001470 016202 MOV L$CHTB(R2),R2
651 001474 022702 000030 CMP #120000,R2
652 001500 101004 120000 BHI $
653 001502 042702 160000 BIC #160000,R2
654 001506 052702 120000 BIS #120000,R2
655 001512 C10267 000000G 5$: MOV R2, HSHADD
656 001516 005002 CLR R2
657 001520 016700 000010' MOV PVCLCN,R0
658 001524 016701 000000G 10$: MOV HSHSZ,R1
659 001530 005401 NEG R1
660 001532 040100 BIC R1,R0
661 001534 010001 MOV R0,R1
662 001536 006300 ASL R0
663 001540 006300 ASL R0
664 001542 066700 000000G ADD HSHADD,R0
665 001546 005760 000002 TST 2(R0)
666 001552 001407 BEQ 20$
667 001554 005202 INC R2
668 001556 026702 000000G CMP HSHSZ,R2
669 001562 103407 BLO 30$
670 001564 060201 ADD R2,R1
671 001566 010100 MOV R1,R0
672 001570 000755 BR 10$
673 001572 016720 000010' 20$: MOV PVCLCN,(R0)+
674 001576 116710 000034' 30$: MOV PVCPRT,(R0)
675 001602 RESMAP
676 001606 RETURN
  
```

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL    | VALUE       | REFERENCES   |
|-----------|-------------|--|
| X\$PKSZ   | 000016      | *17-613  |
| X\$PRT    | 000021      | *17-616  |
| X\$RXQ    | 000072      | *17-620      17-621      *17-621   |
| X\$ST     | 000005      | *17-626  |
| X\$TCLZ   | 000030      | *14-502  |
| X\$TXQ    | 000066      | *17-618      17-619      *17-619   |
| X\$TYP    | 000015      | *17-627  |
| X\$WAQ    | 000062      | *17-622      17-623      *17-623   |
| X\$WSZ    | 0J00020     | *17-612  |
| ZTIME     | = ***** GX  | 14-500   |
| Z.DAT     | = ***** GX  | 21-803   |
| Z.MAP     | = ***** GX  | 21-800   |
| Z.NAM     | = ***** GX  | 21-795   |
| \$ALPHA   | = 000022    | #10-208  |
| \$ANY     | = 000020    | #10-208  |
| \$BLANK   | = 000006    | #10-208  |
| \$CAT5    | = ***** GX  | 12-381      20-712   |
| \$CEACX   | = ***** GX  | 13-427      14-486      15-530      16-569      17-603      18-648                                     |
| \$DIGIT   | = 000024    | #10-208  |
| \$DNUMB   | = 000014    | #10-208  |
| \$EOS     | = 000012    | #10-208  |
| \$ERRN0   | 000052 R    | #8-161      11-319      13-464      20-732   |
| \$ERRN1   | 000112 R    | #8-162      12-387   |
| \$ERRN2   | 000146 R    | #8-163      12-401   |
| \$ERRN3   | 000200 R    | #8-164      13-465   |
| \$ERRQ0   | 000300 R    | #8-166      12-372   |
| \$ERRQ5   | 000334 R    | #8-167      11-347   |
| \$ERRQ9   | 000370 R    | #8-168      11-335   |
| \$ERRP8   | 000244 R    | #8-165      12-360   |
| \$ERR1T   | = ***** GX  | 9-202  |
| \$EXIT    | = 000000    | #10-208  |
| \$FAIL    | = 177777    | #10-208  |
| \$GPRM    | = *****     | 10-208   |
| \$HEADR   | = ***** GX  | 13-423      13-441      13-450      15-544      20-720   |
| \$LAMDA   | = 000000    | #10-208  |
| \$NUMBER  | = 000002    | #10-208  |
| \$QPVC    | = 000000 RG | #9-186   |
| \$RAD50   | = 000016    | #10-208  |
| \$RONLY   | = *****     | 10-208      10-208   |
| \$SLTA    | = ***** GX  | 20-742      21-776   |
| \$STRNG   | = 000004    | #10-208  |
| \$SUBXP   | = 000010    | #10-208  |
| \$TRIB    | = ***** GX  | -750   |
| \$XALOC   | = ***** GX  | -421      14-482   |
| \$X3DBS   | = ***** GX  | 11-311   |
| \$X3DWS   | = ***** GX  | 11-312   |
| \$SHTIM   | = ***** GX  | 13-409   |
| \$SLT.M   | = ***** GX  | 13-410   |
| \$\$\$FLG | 177777      | #10-208  |
| \$\$\$KEY | 177777      | #10-208  |
| .PNUMB    | = ***** GX  | 11-326      11-342      12-394      12-396      20-737      20-740      20-743      20-749      20-750 |

CFGSL

SYMBL

SYMBL

CFGBF

CFGSZ  
CTL

LF-AC  
LF-BE

**THE**

LF . LF  
LF MF

LF . MF  
LE . MT

LF.FA  
LE.BC

LF .RE  
LF .SE

LF . TJ

LF. X2

LN.DU

LN.LC

LN. OF

LN.00

LN. RE

LN. S'

LN. SU  
LN. TA

L. C. T.

L.DDM

L. D. D. S.

L.DL7  
L.DL7

L. FLO  
L. KPR

L. L. L.  
MD

L. NM  
L. NC

LOW

SF-A

SF.L

574

CFGSLT      CREATED BY    MACRO    ON 29-JUN-85 AT 00:05      PAGE 1      D 11

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE    | REFERENCES |
|--------|----------|------------|
| CFGBF  | = *****  | GX 6-88    |
| CFGSZ  | = *****  | GX 6-87    |
| CTL    | = *****  | GX 9-192   |
| LF.ACT | = 100000 | #5-58      |
| LF.BRO | = 000400 | #5-58      |
| LF.BWT | = 000007 | #5-58      |
| LF.ENA | = 002000 | #5-58      |
| LF.LPB | = 001000 | #5-58      |
| LF.MDC | = 000100 | #5-58      |
| LF.MFL | = 004000 | #5-58      |
| LF.MTP | = 000020 | #5-58      |
| LF.FAC | = 000200 | #5-58      |
| LF.RDY | = 040000 | #5-58      |
| LF.REA | = 010000 | #5-58      |
| LF.SER | = 000040 | #5-58      |
| LF.TIM | = 000010 | #5-58      |
| LF.UNL | = 020000 | #5-58      |
| LF.X2P | = 000000 | #5-58      |
| LN.CLO | = 000000 | #5-58      |
| LN.DUM | = 000005 | #5-58      |
| LN.LOA | = 000004 | #5-58      |
| LN.LOO | = 000003 | #5-58      |
| LN.OAU | = 000003 | #5-58      |
| LN.OFF | = 000001 | #5-58      |
| LN.ON  | = 000000 | #5-58      |
| LN.OCF | = 000004 | #5-58      |
| LN.OPE | = 000001 | #5-58      |
| LN.REF | = 000002 | #5-58      |
| LN.SER | = 000002 | #5-58      |
| LN.STA | = 000017 | #5-58      |
| LN.SUB | = 000360 | #5-58      |
| LN.TRI | = 000006 | #5-58      |
| L.COST | = 000015 | #5-58      |
| L.CTL  | = 000012 | #5-58      |
| L.CVA  | = 177776 | #5-58      |
| L.DDM  | = 000002 | #5-58      |
| L.DDS  | = 000004 | #5-58      |
| L.DLC  | = 000003 | #5-58      |
| L.DLM  | = 000002 | #5-58      |
| L.DLS  | = 000010 | #5-58      |
| L.FLG  | = 000000 | #5-58      |
| L.KRBA | = 000016 | #5-58      |
| L.LCN  | = 000022 | #5-58      |
| L.MPF  | = 000022 | #5-58      |
| L.NMST | = 000020 | #5-58      |
| L.NSTA | = 000014 | #5-58      |
| L.OWNR | = 000021 | #5-58      |
| L.UNT  | = 000013 | #5-58      |
| SF.ACT | = 000200 | #5-58      |
| SF.ENA | = 000100 | #5-58      |
| SF.LPB | = 000004 | #5-58      |
| SF.MFL | = 000040 | #5-58      |

CFGSTA - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 00:06 Page 11  
 STA\$DF ACTION ROUTINES

```

229
230
231      ; HELLO TIMER
232
233 000500 005767 000000G  ST.HTM: TST      .PNUMH      ; DOUBLE WORD VALUE?
234 000504 001015          BNE      101$      ; BR IF YES - ERROR
235 000506 016700 000000G  MOV      SCNT,RO    ; GET CURRENT STATION COUNT
236 000512 005300          DEC      RO        ; REALLY LAST ONE
237 000514 006300          ASL      RO        ; MAKE IT A WORD OFFSET
238 000516 016760 000000G  MOV      .PNUMB,$$HTIM(RO) ; SAVE HELLO TIMER
239 000524 006367 000000G  ASL      .PNUMB      ; MULTIPLY BY 2
240 000530 016750 000000G  MOV      .PNUMB,$$LTIM(RO) ; STORE LISTEN TIMER
241 000536          RETURN
242
243 000540      101$:  MSG$R  YA          ; ILLEGAL HELLO TIMER
244
245          000001      .END

```



```

181          .SBTTL SVC$DF STATE TABLE
182          ;
183          ; TPARS STATE TABLES
184          ;
185          ISTAT$ SVCST,SVCKW
186          ;
187          ; SVC$DF
188          ;
189          STATES$ SVCDF
190          TRANS$ %SVC$DF%,...1,SYNERR
191
192          STATES$
193          TRANS$ $RAD50,,SVNAM ; PROCESS NAME
194
195          STATES$
196          TRANS$ '-'
197
198          STATES$
199          TRANS$ $DNUMB,,SVCTL ; CONTROLLER NUMBER
200
201          STATES$
202          TRANS$ '<','>'
203
204          STATES$
205          TRANS$ $DNUMB,,SVSTA ; STATION NUMBER
206
207          STATES$
208          TRANS$ '<','>'
209
210          STATES$
211          TRANS$ '<','>',SVCWND ; PACKET SIZE
212          TRANS$ $NUMBR,,SVPKT
213
214          STATES$
215          TRANS$ '<','>'
216
217          STATES$
218          TRANS$ SVCWND ; WINDOW SIZE
219          TRANS$ '<','>',SVCFLG
220          TRANS$ $NUMBR,,SVWND
221
222          STATES$
223          TRANS$ '<','>'
224
225          STATES$
226          TRANS$ SVCFLG ; FLAGS
227          TRANS$ $NUMBR,,SVFLAG
228
229          STATES$
230          TRANS$ '<','>'
231
232          STATES$
233          TRANS$ $NUMBR,,SVRCLL ; RECALL TIMER
234
235          STATES$
236          TRANS$ '<','>'
237
238          STATES$
239          TRANS$ $NUMBR,,SVRTRY ; RETRY TIMER

```

|                    |                  |                 |                     |                       |
|--------------------|------------------|-----------------|---------------------|-----------------------|
| SF.MFL= 000040     | SYNERR= ***** GX | T\$LLD 000012   | V\$LEN 000022       | \$ERR3Z 000374R 002   |
| SF.PAC= 000020     | \$SADSZ 000007   | T\$LLDC 000045  | V\$RCV 000002       | \$EXIT = 000000       |
| SF.REA= 000010     | \$SCTL 000002    | T\$LLDL 000012  | V\$XMT 000012       | \$FAIL = 177777       |
| SF.SER= 000001     | \$SDTE 000020    | T\$LLDO 000012  | V\$SCTR= 001000     | \$HEADR= ***** GX     |
| SF.SVC= 000002     | \$SFLG 000010    | T\$LLDS 000012  | WNDSVC 000022R 002  | \$LAMBDA= 000000      |
| SF.UNL= 000040     | \$SLEN 000032    | T\$LLEN 000046  | X\$SDBT= 000000     | \$LINKX= ***** GX     |
| SLTA 000016R 002   | \$SLFT 000012    | T\$LOPR 000002  | Z.DAT = ***** GX    | \$NUMBR= 000002       |
| SLTMA = ***** GX   | \$SLNK 000000    | T\$LTCL 000024  | Z.MAP = ***** GX    | \$QSV 000000RG        |
| STASVC 000014R 002 | \$SOWNR 000030   | T\$LTIM 000026  | \$ALPHA= 000022     | \$RAD50= 000016       |
| STRNXT 001520R     | \$SPKSZ 000004   | T\$LTFR 000014  | \$ANY = 000020      | \$SAVRG= ***** GX     |
| SVCDF 000000R      | \$SRCL 000013    | T\$LTFS 000020  | \$BLANK= 000006     | \$SLTA = ***** GX     |
| SVCDFE 000110R     | \$SSTA 000003    | T\$SNAPL 000004 | \$CAT5 = ***** GX   | \$STRNG= 000004       |
| SVCFLG 000054R     | \$STIM 000014    | T\$NFE 000000   | \$CDTB = ***** GX   | \$SUBXP= 000010       |
| SVCCKW 000000RG    | \$STMR 000016    | T\$NLEN 000010  | \$CEACX= ***** GX   | \$TRIB = ***** GX     |
| SVCST 000000RG     | \$SWND 000006    | T\$NNUL 000002  | \$DIGIT= 000024     | \$XALOC= ***** GX     |
| SVCCTL 000114R     | \$S\$VRG= 000000 | T\$NOPL 000006  | \$DNUMB= 000014     | \$SHTIM= ***** GX     |
| SVCWND 000042R     | \$S\$VSZ= 007600 | T\$NNRNI 000042 | \$EOS = 000012      | \$SLTIM= ***** GX     |
| SVDTEd 000476R     | S.COST 000001    | T\$NRPL 000005  | \$ERR1T= ***** GX   | \$SFLG= 177777        |
| SVDTEE 000522R     | S.FLG 000000     | T\$NRUL 000007  | \$ERR2I= ***** GX   | \$SKEY= 000000        |
| SVDTES 000440RG    | S.LEN 000004     | T\$NVR 000001   | \$ERR3D 000056R 002 | \$S\$R = 000010       |
| SVFLAG 000316R     | S.NMST 000002    | T\$RPR1 000040  | \$ERR3E 000102R 002 | \$S\$STA= 000000      |
| SVNAM 000052R      | S.OWNR 000003    | T\$SVC 000034   | \$ERR3F 000126R 002 | \$S\$TMP= 000000R 005 |
| SVOWN 000122R 003  | T\$FLAG 000044   | T\$T5 000030    | \$ERR3G 000152R 002 | .PCHAR= ***** GX      |
| SVOWNR 000550R     | T\$LIF 000013    | T\$T6 000032    | \$ERR3H 000200R 002 | .PNUMB= ***** GX      |
| SVPKT 000236R      | T\$LIFL 000013   | T\$SKMG= 000000 | \$ERR3I 000224R 002 | .PNUMH= ***** GX      |
| SVRCLL 000342R     | T\$LIFO 000013   | T\$SMIN= 000000 | \$ERR3J 000250R 002 | .PSTCN= ***** GX      |
| SVRTRY 000374R     | T\$LIFS 000013   | V\$SRCV= 100000 | \$ERR3K 000266R 002 | .PSTPT= ***** GX      |
| SVSTA 000164R      | T\$LIN 000000    | V\$XMT= 040000  | \$ERR3M 000340R 002 | .TPARS= ***** GX      |
| SVWND 000262R      | T\$LIPS 000006   | V\$FLG 000000   |                     |                       |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
001554 001 (RW,I,LCL,REL,CON)  
DATA 000424 002 (RW,D,LCL,REL,CON)  
\$STATE 000130 003 (RW,D,LCL,REL,CON)  
\$KTAB 000002 004 (RW,D,LCL,REL,CON)  
\$KSTR 000007 005 (RW,D,LCL,REL,CON)  
Errors detected: 0

### \*\*\* Assembler statistics

Work file reads: 55  
Work file writes: 61  
Size of work file: 20001 Words ( 79 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:53.28  
SY:CFG SVC.V2,[132,134]CFG SVC/CR/-SP=SY:[1,1]RSXCMC.SML/ML,[130,110]NETLIB/ML,[130,10]RSXCMC/PA:1,[132,10]CFG SVC

```

121                                     .SBTTL  $QUNT - LOOK FOR UNT$DF MACRO
122
123                                     ;+
124                                     ;
125                                     ; $QUNT - LOOK FOR UNT$DF MACRO
126                                     ;
127                                     ; INPUTS:
128                                     ;     NONE
129                                     ;
130                                     ; OUTPUTS:
131                                     ;     CARRY SET - UNT$DF NOT FOUND
132                                     ;     CARRY CLEAR - UNT$DF FOUND
133                                     ;
134                                     ; -
135 000040 012705 000000' $QUNT:: MOV      #UNTDF,R5      ; STATE TABLE ADDRESS
136 000044 005001          CLR      R1              ; FULL KEYWORD MATCH LENGTH
137 000046 012702 000000' MOV      #UNTKW,R2      ; KEYWORD TABLE ADDRESS
138 000052 016703 000000G MOV      CFGSZ,R3      ; RECORD LENGTH
139 000056 012704 000000G MOV      #CFGBF,R4      ; RECORD BUFFER ADDRESS
140 000062 005067 000000G CLR      SYNERR      ; CLEAR SYNTAX ERROR FLAG
141 000066          CALL    TPARS      ; GO DO THE PARSE
142 000072 103003          BCC      20$      ; IF CC, FOUND WHAT WE WERE LOOKING FOR
143 000074 005367 000000G BEQ      SYNERR      ; DID SYNTAX ERROR OCCUR?
144 000100 001401          BEQ      101$      ; IF EQ, YES
145 000102          20$:  RETURN
146
147          ; ERRORS
148
149 000104          101$:  MSG$R  1T          ; SYNTAX ERROR
    
```

CFGUNT      CREATED BY    MACRO    ON 15-JUL-85 AT 18:57      PAGE 3      D 16

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL   | VALUE       | REFERENCES  |
|----------|-------------|---|
| \$ANY    | = 000020    | #9-153  |
| \$BLANK  | = 000006    | #9-153  |
| \$DIGIT  | = 000024    | #9-153  |
| \$DIV    | = ***** GX  | 12-309 12-312   |
| \$DNUMB  | = 000014    | #9-153  |
| \$EOS    | = 000012    | #9-153  |
| \$ERRYP  | = ***** GX  | 13-400  |
| \$ERRZO  | = 000254 R  | #7-116 13-365   |
| \$ERRIG  | = 000000 R  | #7-110 11-222   |
| \$ERRIH  | = 000024 R  | #7-111 11-223   |
| \$ERRIK  | = 000060 R  | #7-112 11-246   |
| \$ERRIL  | = 000116 R  | #7-113 13-363   |
| \$ERRIM  | = 000154 R  | #7-114 13-364   |
| \$ERRIN  | = 000216 R  | #7-115 13-378   |
| \$ERRIT  | = ***** GX  | 8-149   |
| \$EXIT   | = 000000    | #9-153  |
| \$FAIL   | = 177777    | #9-153  |
| \$FLAGS  | = ***** GX  | 11-235 11-238 13-339 13-342 13-370 13-372                       |
| \$GPRM   | = *****     | 9-153   |
| \$LAMDA  | = 000000    | #9-153  |
| \$MISS   | = ***** GX  | 13-352  |
| \$NLCSR  | = ***** GX  | 13-358  |
| \$NUMBER | = 000002    | #9-153  |
| \$QUNT   | = 000040 RG | #8-135  |
| \$RAD50  | = 000016    | #9-153  |
| \$RONLY  | = *****     | 9-153 9-153 9-153 9-153 9-153                                   |
| \$SLTA   | = ***** GX  | 11-231 11-255 11-270 12-302 13-351 13-385                       |
| \$STRNG  | = 000004    | #9-153  |
| \$SUBXP  | = 000010    | #9-153  |
| \$SDCHA  | = ***** GX  | *11-241 *11-258 *11-261 12-291 12-295 12-306                    |
| \$SDPR   | = ***** GX  | *13-406   |
| \$SPCHA  | = ***** GX  | *11-273 *11-275   |
| \$SSCSR  | = ***** GX  | *13-360   |
| \$SFLG   | = 177777    | #9-153  |
| \$SKEY   | = 177777    | #9-153  |
| .PNUMB   | = ***** GX  | 11-217 11-219 11-232 11-258 11-261 11-273 11-275 12-285 *12-331 |
| .PNUMH   | = ***** GX  | *12-332 13-383 13-395 13-397 13-406                             |
| .TPARS   | = ***** GX  | 11-215 11-228 11-251 11-267                                     |
| ..SCSR   | = ***** GX  | 8-141 13-350  |

EVL            CREATED BY    MACRO    ON 28-JUN-85 AT 22:58            PAGE 6            E 1

SYMBOL CROSS REFERENCE            CREF    04.CO

| SYMBOL  | VALUE      | REFERENCES               |
|---------|------------|--------------------------|
| SC\$RST | = 000003   | #5-61                    |
| SC\$SHU | = 000002   | #5-61                    |
| SF.ACT  | = 000200   | #5-60                    |
| SF.ENA  | = 000100   | #5-60                    |
| SF.LPB  | = 000004   | #5-60                    |
| SF.MFL  | = 000040   | #5-60                    |
| SF.PAC  | = 000020   | #5-60                    |
| SF.REA  | = 000010   | #5-60                    |
| SF.SER  | = 000001   | #5-60                    |
| SF.SVC  | = 000002   | #5-60                    |
| SF.UNL  | = 000040   | #5-60                    |
| SLTMA   | = 000036 R | #8-111            13-459 |
| SV\$DUM | = 000001   | #5-61                    |
| SV\$LOA | = 000000   | #5-61                    |
| S.COST  | = 000001   | #5-60                    |
| S.FLG   | = 000000   | #5-60                    |
| S.LEN   | = 000004   | #5-60                    |
| S.NMST  | = 000002   | #5-60                    |
| S.OWNR  | = 000003   | #5-60                    |
| TTNS    | = 000012 R | #8-94            14-494  |
| XT\$PVC | = 000001   | 13-440                   |
| X\$TYP  | = 000015   | 13-440                   |
| X\$USR  | = 000022   | 13-437                   |
| X\$MBCB | = *****    | 5-62                     |
| ZERTAB  | = 000422 R | 9-247            #9-275  |
| Z.CO    | = 001000   | #5-62                    |
| ZF.DDM  | = 000001   | #5-62                    |
| ZF.DIA  | = 004000   | #5-62                    |
| ZF.DLC  | = 000002   | #5-62                    |
| ZF.DVP  | = 100000   | #5-62                    |
| ZF.INI  | = 040000   | #5-62                    |
| ZF.KMX  | = 000020   | #5-62                    |
| ZF.LLC  | = 000004   | #5-62                    |
| ZF.LMC  | = 000100   | #5-62                    |
| ZF.MAN  | = 020000   | #5-62                    |
| ZF.MFL  | = 000010   | #5-62                    |
| ZF.MTM  | = 000400   | #5-62                    |
| ZF.MUX  | = 000040   | #5-62                    |
| ZF.PSE  | = 002000   | #5-62                    |
| ZF.SLI  | = 010000   | #5-62                    |
| ZF.TIM  | = 000200   | #5-62                    |
| ZF.X3P  | = 000000   | #5-62                    |
| ZS.ASN  | = 100000   | #5-62                    |
| ZS.BSY  | = 140000   | #5-62                    |
| ZTIM2   | = 000040 R | #8-112            9-265  |
| Z.AVL   | = 000014   | #5-62                    |
| Z.DAT   | = 000016   | #5-62            10-301  |
| Z.DSP   | = 000000   | #5-62            5-62    |
| Z.FLG   | = 000010   | #5-62                    |
| Z.LEN   | = 000016   | #5-62                    |
| Z.LLN   | = 000006   | #5-62                    |
| Z.MAP   | = 000020   | #5-62            12-404  |

EVL            CREATED BY    MACRO    ON 28-JUN-85 AT 22:58            PAGE 7            F 1

```

343 000330 061216      ADD      (R2),(SP)      ; = BASE + SIZE
344 000332 024226      CMP      -(R2),(SP)+    ; IS NEXT HOLE POINTER (BASE
345                                     ; ADDRESS OF UPPER BOUND) = TO
346 000334 001007      BNE      80$           ; LOW ADDR + LOW SIZE?
347 000336 011500      MOV      (R5),R0        ; YES -- COMPACT LOWER AND UPPER
348                                     ; BOUNDS. SAVE LOW BASE ADDR.
349 000340 011215      MOV      (R2),(R5)      ; SET ASR TO ACCESS UPPER
350                                     ; BOUND ARGS.
351 000342 012246      MOV      (R2)+,-(SP)    ; = NEXT HOLE THAT UPPER BOUND
352                                     ; POINTS TO.
353 000344 011246      MOV      (R2),-(SP)     ; = UPPER BOUND SIZE.
354 000346 010015      MOV      R0,(R5)       ; RESET ASR TO LOW BOUND HOLE.
355 000350 062612      ADD      (SP)+,(R2)     ; LOW SIZE = OLD LOW SIZE + UPPER
356                                     ; BOUND HOLE SIZE
357
358 000352 012642      MOV      (SP)+,-(R2)    ; LOWER BOUND NOW POINTS TO
359                                     ; HOLE THAT UPPER BOUND HOLE
360                                     ; USED TO AND UPPER BOUND
361                                     ; HOLE IS GONE.
362 000354 005726      80$: TST      (SP)+      ; CLEAN STACK
363 000356                                     ;
364
365 000356 012677 177416 MOV      (SP)+,aKS6    ; RESTORE MAPPING REGISTER
366 000362      RETURN
367
368      000001      .END
  
```

\*\*FILE\*\*ID\*\*CFGENT

E 3

```
CCCCCCCC FFFFFFFF GGGGGGGG CCCCCCCC NN NN TTTTTTTTTT
CCCCCCCC FFFFFFFF GGGGGGGG CCCCCCCC NN NN TTTTTTTTTT
CC FF GG CC NN NN TT
CC FF GG CC NN NN TT
CC FF GG CC NNNN NN TT
CC FF GG CC NNNN NN TT
CC FFFFFFFF GG CC NN NN TT
CC FFFFFFFF GG CC NN NN TT
CC FF GG GGGGGG CC NN NNNN TT
CC FF GG GGGGGG CC NN NNNN TT
CC FF GG GG CC NN TT
CC FF GG GG CC NN TT
CC FF GG GG CC NN TT
CCCCCCCC FF GGGGGG CCCCCCCC NN NN TT
CCCCCCCC FF GGGGGG CCCCCCCC NN NN TT
```

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

\*\*FILE\*\*ID\*\*CFGDDM

```

CCCCCCCC FFFFFFFF GGGGGGGG DDDDDDDD DDDDDDDD MM MM
CCCCCCCC FFFFFFFF GGGGGGGG DDDDDDDD DDDDDDDD MM MM
CC FF GG DD DD MM MM
CC FF GG DD DD MMMM MMMM
CC FF GG DD DD MMMM MMMM
CC FF GG DD DD MM MM
CC FFFFFFFF GG DD DD MM MM
CC FFFFFFFF GG DD DD MM MM
CC FF GGGGGG DD DD MM MM
CC FF GGGGGG DD DD MM MM
CC FF GG GG DD DD MM MM
CC FF GG GG DD DD MM MM
CCCCCCCC FF GGGGGG DDDDDDDD DDDDDDDD MM MM
CCCCCCCC FF GGGGGG DDDDDDDD DDDDDDDD MM MM

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SS:SSS TT
LL SS:SSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52

.TITLE CFGDTE - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 28-OCT-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

BIASX - SET APR6 BIAS CORRECTLY

.SBTTL BIASX - SET APR6 BIAS CORRECTLY

395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414

000500  
000500 020027 120000  
000504 103404  
000506 042700 160000  
000512 052700 140000  
000516

```

: +
: BIASX - SET APR6 BIAS CORRECTLY
: INPUTS:
:   R0 = ADDRESS (POOL OR APR5)
: OUTPUTS:
:   R0 = ADDRESS (POOL OR APR5)
: -
BIASX:  CMP    R0,#120000      ;; ADDRESS IN APR5 ?
        BLO    10$           ;; BR IF NO (ASSUME POOL)
        BIC    #160000,R0    ;; CLEAR ALL APR BITS
        BIS    #140000,R0    ;; AND BIAS FOR APR6
10$:   RETURN                ;; RETURN

```

55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71

.SBTTL MACRO DEFINITIONS

;; LIBRARY MACROS

;; .MCALL ISTAT\$,TRAN\$,STAT\$,EMSG\$R

;; LOCAL MACRO DEFINITIONS

;; REJECT TPARS TRANSITION

;; .MACRO REJS  
ADD #2,(SP) ; RETURN TO CALLER+2  
CLR SYNER ; INDICATE NO SYNTAX ERROR  
;; .ENDM REJS

CFGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:04 Page 8  
 ERROR MESSAGES

```

157 .SBTTL ERROR MESSAGES
158
159 .ENABL LC
160
161 NTLR$ ,N0,8B,CERR,RTSPC,CFLIN,<circuit identification>
162 NTLR$ ,N1,8B,CERR,RTSPC,CFLIN,<process owner name>
163 NTLR$ ,N2,8B,CERR,RTSPC,CFLIN,<flags byte value>
164 NTLR$ ,N3,8,CERR,RTSPC,CFLIN,<Resource allocation failure>
165 NTLR$ ,P8,8B,CERR,RTSPC,CFLIN,<maximum block size>
166 NTLR$ ,O0,8B,CERR,RTSPC,CFLIN,<maximum window size>
167 NTLR$ ,O5,8B,CERR,RTSPC,CFLIN,<counter timer value>
168 NTLR$ ,O9,8B,CERR,RTSPC,CFLIN,<logical channel number>
169 .DSABL LC
170 .EVEN
171 .LIST BEX
172 .PSECT
172 000000

```

CFGP  
 BIAS

BIASX - BIAS XPOOL ADDRESS FOR APR6

```

678 .SBTTL BIASX - BIAS XPOOL ADDRESS FOR APR6
679
680
681 ;+
682 ; BIASX - BIAS XPOOL ADDRESS FOR APR6
683
684 ; INPUTS:
685 ; R0 = ADDRESS (POOL OR APR5)
686
687 ; OUTPUTS:
688 ; R0 = ADDRESS (POOL OR APR6)
689 ; -
690
691 BIASX:
692 001610 020027 120000    CMP     R0,#120000    ; POOL ADDRESS?
693 001614 103404          BLO     10$           ; YES, LEAVE AS IS
694 001616 042700 160000   BIC     #160000,R0    ; CLEAR APR BITS
695 001622 052700 140000   BIS     #140000,R0    ; BIAS FOR APR6
696 001626          10$:    RETURN                ; RETURN

```

CFGPVC      CREATED BY    MACRO    ON 29-JUN-85 AT 00:04      PAGE 6      E 10

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE            | REFERENCES                                     |
|--------|------------------|--|
| .PNUMH | =    *****    GX | 11-324    11-340    12-392    20-738    20-752 |
| .PSTCN | =    *****    GX | 11-298    12-377                               |
| .PSTPT | =    *****    GX | 11-306    12-379                               |
| .TPARS | =    *****    GX | 9-193  |

CFGSL  
SYMBBC  
SYMBBC  
GF.PA  
SF.RE  
SF.SE  
SF.SV  
SF.UN  
SLTDF  
SLTKW  
SLTST  
SYNER  
S.BRA  
S.COS  
S.CTL  
S.CTL  
S.FLC  
S.HT  
S.LEN  
S.NMS  
S.OW  
S.RPF  
S.UN  
UNT  
\$ALPH  
\$ANY  
\$BLA  
\$CTI  
\$DIG  
\$DNUI  
\$EOS  
\$ERR  
\$ERR  
\$ERR  
\$EXI  
\$FAI  
\$GPRI  
\$LAMI  
\$NUM  
\$OSL  
\$RAD  
\$RON  
\$STR  
\$SUB  
\$HT  
\$LT  
\$SF  
\$SK  
NBR  
PNU  
RPR  
TPA

| SYMBOL    | VALUE      | REFERENCES                 |
|-----------|------------|----------------------------|
| SF.PAC    | = 000020   | #5-58                      |
| SF.REA    | = 000010   | #5-58                      |
| SF.SER    | = 000001   | #5-58                      |
| SF.SVC    | = 000002   | #5-58                      |
| SF.UNL    | = 000040   | #5-58                      |
| SLTDF     | 000000 R   | 6-84                       |
| SLTKW     | 000000 RG  | 6-86                       |
| SLTST     | 000000 RG  | #7-107                     |
| SYNERR    | = ***** GX | *6-94 *9-194 *9-202 *9-241 |
| S.BRAD    | = 000210 R | #9-229                     |
| S.COST    | 000001     | #5-58                      |
| S.CTIM    | 000126 R   | #9-207                     |
| S.CTL     | 000062 R   | #9-192                     |
| S.FLG     | 000000     | #5-58                      |
| S.HTIM    | 000152 R   | #9-216                     |
| S.LEN     | 000004     | #5-58                      |
| S.NMST    | 000002     | #5-58                      |
| S.OWNR    | 000003     | #5-58                      |
| S.RPRI    | 000220 R   | #9-236                     |
| S.UNT     | 000104 R   | #9-200                     |
| UNT       | = ***** GX | 9-200                      |
| \$ALPHA   | = 000022   | #7-107                     |
| \$ANY     | = 000020   | #7-107                     |
| \$BLANK   | = 000006   | #7-107                     |
| \$CTIM    | = ***** GX | *9-209                     |
| \$DIGIT   | = 000024   | #7-107                     |
| \$DNUMB   | = 000014   | #7-107                     |
| \$EOS     | = 000012   | #7-107                     |
| \$ERRYR   | = ***** GX | 9-211                      |
| \$ERRZP   | = ***** GX | 9-223                      |
| \$ERRIT   | = ***** GX | 6-101                      |
| \$EXIT    | = 000000   | #7-107                     |
| \$FAIL    | = 177777   | #7-107                     |
| \$GPRM    | = *****    | 7-107                      |
| \$LAMBDA  | = 000000   | #7-107                     |
| \$NUMBR   | = 000002   | #7-107                     |
| \$QSLT    | 000000 RG  | #6-84                      |
| \$RAD50   | = 000016   | #7-107                     |
| \$RONLY   | = *****    | 7-107                      |
| \$STRNG   | = 000004   | #7-107                     |
| \$SUBXP   | = 000010   | #7-107                     |
| \$SHTIM   | = ***** GX | *9-218                     |
| \$SLTIM   | = ***** GX | *9-220                     |
| \$\$\$FLG | = 177777   | #7-107                     |
| \$\$\$KEY | = 177777   | #7-107                     |
| \$.NBRA   | = ***** GX | *6-90 *9-229               |
| \$.PNUMB  | = ***** GX | 9-192 9-200                |
| \$.PNUMH  | = ***** GX | 9-207 9-216                |
| \$.RPRI   | = ***** GX | *6-91 *9-238               |
| \$.TPAPS  | = ***** GX | 6-92                       |

A\$\$\$C  
 A\$\$\$C  
 A\$\$\$P  
 A\$\$\$T  
 BITS  
 BITS  
 BITS  
 CERR  
 CFGB  
 CFGS  
 CFLI  
 C\$\$\$C  
 C\$\$\$C  
 C\$\$\$R  
 D\$\$\$E  
 D\$\$\$I  
 D\$\$\$L  
 D\$\$\$Y  
 D\$\$\$Y  
 END  
 E\$\$\$X  
 FMTE  
 FMTE  
 FM.£  
 FM.£  
 F\$\$\$L  
 G\$\$\$1  
 G\$\$\$1  
 G\$\$\$1  
 G\$\$\$W  
 I\$\$\$F  
 I\$\$\$F  
 K\$\$\$G  
 K\$\$\$G  
 K\$\$\$I

. AE  
 \$STA  
 \$KTA  
 \$KS  
 Error

\*\*\*

|      |  |
|------|--|
| Work |  |
| Work |  |
| Size |  |
| Size |  |
| Open |  |

ELa  
SY:

## Symbol table

```

ASSCHK= 000000      KSSIPS= 000074      L.COST 000015      RSSDER= 000000      $ALPHA= 000022
ASSCPS= 000000      LD$LP = 000000      L.CTL  000012      RSSK11= 000001      $ANY = 000020
ASSPRI= 000000      LF.ACT= 100000      L.CVA  177776      RSSND= 000000      $BLANK= 000006
ASSTRP= 000000      LF.BRO= 000400      L.DDM  000002      RSS11M= 000000      $DIGIT= 000024
BITS   000076R      002 LF.BWT= 000007      L.DDS  000004      SCNT = ***** GX      $DNUMB= 000014
BITS1   000106R      002 LF.ENA= 002000      L.DLC  000003      SF.ACT= 000200      $EOS = 000012
BITS2   000122R      002 LF.LPB= 001000      L.DLM  000006      SF.ENA= 000100      $ERRYA 000130R
CERR = ***** GX      LF.MDC= 000100      L.DLS  000010      SF.LPB= 000004      $ERRYB 000162R
CFGBF = ***** GX      LF.MFL= 004000      L.FLG  000000      SF.MFL= 000040      $ERRYV= ***** GX
CFGSZ = ***** GX      LF.MTP= 000020      L.KRBA 000016      SF.PAC= 000020      $ERRYQ 000070R
CFLIN = ***** GX      LF.PAC= 000200      L.LEN = 000022      SF.REA= 000010      $ERRIA 000000R
C$CKP= 000000      LF.RDY= 040000      L.MPF  000022      SF.SER= 000001      $ERRIB 000030R
C$SORE= 000400      LF.REA= 010000      L.NMST 000020      SF.SVC= 000002      $ERRIT= ***** GX
C$RSH= 177564      LF.SER= 000040      L.NSTA 000014      SF.UNL= 000040      $EXIT = 000000
D$BUG= 177514      LF.TIM= 000010      L.OWNR 000021      STADF 000000R      002 $FAIL = 177777
D$ISK= 000000      LF.UNL= 020000      L.UNIT 000013      STADF2 000032R      002 $LAMBDA= 000000
D$SL11= 000001      LF.X2P= 000000      MAXCST= 000031      STAKW 000000RG      003 $NUMBR= 000002
D$SYNC= 000000      LN.CLD= 000000      M$CRB= 000124      STAST 000000RG      002 $QSTA 000216RG
D$SYNM= 000000      LN.DUM= 000005      M$CRX= 000000      STAPR 000446R      $RAD50= 000016
END      000066R      002 LN.LDA= 000004      M$FCS= 000000      ST.CST 000356R      $SLTA = ***** GX
E$XPR= 000000      LN.LOO= 000003      M$HGE= 000000      ST.HTM 000500R      $STRNG= 000004
FMT8 = ***** GX      LN.OAU= 000003      M$NET= 000000      ST.NUM 000270R      $SUBXP= 000010
FMT8B = ***** GX      LN.OFF= 000001      M$OVR= 000000      SYNERR= ***** GX
FM.8 = 000000      LN.ON = 000000      M$ACC= 000001      S$WRG= 000000      $$APR = ***** GX
FM.8B = 000000      LN.OOP= 000004      M$BUF= 000001      S$YSZ= 007600      $HITM= ***** GX
F$SLVL= 000001      LN.OPE= 000001      M$LDV= 000001      S.COST 000001      $LITM= ***** GX
G$STPP= 000000      LN.REF= 000002      N$MCP= 000001      S.FLG  000000      $MTP = ***** GX
G$STSS= 000000      LN.SER= 000002      N$MML= 000001      S.LEN  000004      $SSNUM= ***** GX
G$STTK= 000000      LN.STA= 000017      N$MOV= 000010      S.NMST 000002      $SFLG= 177777
G$WRD= 000000      LN.SUB= 000360      N$NCT= 000001      S.OWNR 000003      $SKEY= 000000
I$SRAR= 000000      LN.TRI= 000006      N$SPM= 000001      T$KMG= 000000      $SSTA= 000106R      002
I$SRDN= 000000      L$ASG= 000000      P$P4S= 000000      T$MIN= 000000      $STMP= 000000R      004
K$CNT= 177546      L$DRV= 000000      P$WRD= 000000      V$CTR= 001000      .PNUMB= ***** GX
K$CSR= 177546      L$P11= 000001      Q$OPT= 000010      X$DBT= 000000      .PNUMH= ***** GX
K$LDC= 000000      L$11R= 000000      RTSPC = ***** GX      .TPARS= ***** GX

```

```

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)
      000546 001 (RW,I,LCL,REL,CON)
$STATE 000126 002 (RW,D,LCL,REL,CON)
$KTAB 000002 003 (RW,D,LCL,REL,CON)
$KSTR 000007 004 (RW,D,LCL,REL,CON)

```

Errors detected: 0

## \*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 14408 Words ( 57 Pages)
Size of core pool: 15496 Words ( 59 Pages)
Operating system: RSX-11M/PLUS

```

Elapsed time: 00:00:34.01

SY:CFGSTA.V2,[132,134]CFGSTA/CR/-SP=SY:[1,1]RSXMC.M.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMC/PA:1,[132,10]CFGSTA



CFG SVC - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 00:06 Page 9-1  
 SVC\$DF STATE TABLE

|     |        |  |
|-----|--------|--|
| 238 |        |  |
| 239 | 000052 | STATES   |
| 240 | 000052 | TRANS <','>  |
| 241 |        |  |
| 242 | 000052 | STATES ; START OF DTE ADDRESS                          |
| 243 | 000052 | TRANS <','>,SVOWN,NODTE ; NO DTE ADDRESS SPECIFIED     |
| 244 | 000052 | TRANS \$DIGIT,,SVDTE                                   |
| 245 |        |  |
| 246 | 000052 | STATES SVCDTE  |
| 247 | 000052 | TRANS \$DIGIT,SVCDTE,SVDTE ; NEXT DIGIT IN DTE ADDRESS |
| 248 | 000052 | TRANS <','>,,SVDTEE ; END OF DTE ADDRESS               |
| 249 |        |  |
| 250 | 000052 | STATES SVOWN ; OWNER                                   |
| 251 | 000052 | TRANS \$RAD50,\$EXIT,SVOWNR                            |
| 252 |        |  |
| 253 | 000052 | STATES   |

CFGSVC      CREATED BY    MACRO    ON 29-JUN-85 AT 00:07      PAGE 1      E 14

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES  |
|--------|------------|---|
| CFERR  | = ***** GX | 7-138      7-139      7-140      7-141      7-142      7-143      7-144      7-145      7-146 |
| CFGBF  | = ***** GX | 7-147   |
| CFGSSZ | = ***** GX | 8-169   |
| COUNT  | 000024 R   | 8-168   |
| CTLSVC | 000021 R   | *12-387      13-460      16-602      *17-642  |
| CVTBUF | 000054 R   | *10-279      13-453   |
| DTEEND | = 000044 R | *17-637      17-638   |
| DTELEN | = 000017 R | 12-395  |
| DTEPCK | 000044 R   | 6-129      6-131      12-382      12-408      13-461  |
| DTESVC | 000025 R   | 12-410      13-464  |
| ESNBR  | 000014     | 12-383      12-388      12-409  |
| ESNBS  | 000020     |   |
| ESNCR  | 000034     |   |
| ESNCS  | 000036     |   |
| ESNIC  | 000044     |   |
| ESNLEN | 000050     |   |
| ESNLLA | 000012     |   |
| ESNLNK | 000000     |   |
| ESNML  | 000040     |   |
| ESNMR  | 000024     |   |
| ESNMS  | 000030     |   |
| ESNNOD | 000002     |   |
| ESNRT  | 000042     |   |
| ESNRTP | 000005     |   |
| ESNSEG | 000010     |   |
| ESNTIM | 000046     |   |
| ESNUSE | 000004     |   |
| ESSTRT | 000006     |   |
| FLGSVC | 000010 R   | #5-62   |
| FMT8   | = ***** GX | #6-119      *11-336      12-372      13-456   |
| FMT8B  | = ***** GX | 7-145      7-146      7-147   |
| FM.8   | = 000000   | 7-138      7-139      7-140      7-141      7-142      7-143      7-144                       |
| FM.8B  | = 000000   | #7-145      #7-146      #7-147  |
| FNDOWN | 001146 R   | #7-138      #7-139      #7-140      #7-141      #7-142      #7-143      #7-144                |
| HTIMR  | 000002 R   | 13-424      #14-511   |
| HSSVC  | 000036     | *13-429      15-580   |
| ISSAS  | = *****    | 13-469  |
| KSARS  | = ***** GX | 16-612  |
| LF.ACT | = 100000   | 13-442      13-448      13-483      15-559      15-561      15-582                            |
| LF.BRO | = 000400   |   |
| LF.BWT | = 000007   |   |
| LF.ENA | = 002000   |   |
| LF.LPB | = 001000   |   |
| LF.MDC | = 000100   |   |
| LF.MFL | = 004000   |   |
| LF.MTP | = 000020   |   |
| LF.PAC | = 000200   |   |
| LF.RDY | = 040000   |   |
| LF.REA | = 010000   |   |
| LF.SER | = 000040   |   |
| LF.TIM | = 000010   |   |

```

151 .SBTTL UNT$DF STATE TABLE
152
153 ISTAT$ UNTST,UNTKW
154
155 ; UNIT DEFINITION (UNT$DF)
156 ;
157 STATES UNTDF
158 TRANS %UNT$DF%,1,SYNERR
159 STATES ; UNIT NUMBER
160 TRANS $NUMBR,,U.UNT
161 STATES
162 TRANS <','>
163 STATES ; CHARACTERISTICS WORD 0
164 TRANS $NUMBR,,U.CHA0
165 STATES
166 TRANS <','>
167 STATES ; CHARACTERISTICS WORD 1
168 TRANS $NUMBR,,U.CHA1
169 STATES
170 TRANS !END,$EXIT,U.XCSR
171 TRANS <','>
172
173 STATES ; SECONDARY CSR (OPTIONAL)
174 TRANS $NUMBR,COST,U.SCSR
175 TRANS $LAMDA,,U.XCSR
176
177 STATES COST
178 TRANS <','>,COST1
179 TRANS $LAMDA,$EXIT ; DON'T REQUIRE ANYMORE
180
181 STATES COST1 ; LINE COST
182 TRANS $NUMBR,DPR,U.CST
183 TRANS $LAMDA
184
185 STATES DPR
186 TRANS <','>,DPR1
187 TRANS $LAMDA,$EXIT
188
189 STATES DPR1 ; DEAD POLLING RATIO
190 TRANS $NUMBR,PECH,U.DPR
191 TRANS $LAMDA
192
193 STATES PECH
194 TRANS <','>,PECH1
195 TRANS $LAMDA,$EXIT
196
197 STATES PECH1 ; PROTOCOL EMULATOR CHARACTERISTICS
198 TRANS $NUMBR,$EXIT,U.PECH
199

```

CFGUNT CREATED BY MACRO ON 15-JUL-85 AT 18:57 PAGE 4

MACRO CROSS REFERENCE

CREF 04.00

| MACRO NAME | REFERENCES  |
|------------|---|
| CALL       | 8-141 11-254 12-309 12-312 13-354 13-358                              |
| CHADF\$    | #5-57 5-59  |
| DBGTP\$    | #9-153 #9-171 #9-175 #9-179 #9-183 #9-187 #9-191 #9-195 #10-206       |
| EMSG\$R    | #5-57 8-149 11-222 11-223 11-246 13-363 13-364 13-365 13-378 13-400   |
| ISTAT\$    | #5-57 9-153   |
| MTRANS     | #9-153  |
| NTLR\$     | #5-57 7-110 7-111 7-112 7-113 7-114 7-115 7-116                       |
| RESRG      | #5-57 12-333  |
| RETURN     | 8-145 11-221 11-245 11-262 11-276 12-334 13-362 13-377 13-398 13-407  |
| SAVRG      | #5-57 12-284  |
| SLTDF\$    | #5-57 5-60  |
| STATE\$    | #5-57 9-157 #9-159 #9-161 #9-163 #9-165 #9-167 #9-169 #9-173 #9-177   |
|            | #9-181 #9-185 #9-189 #9-193 #9-197 #10-204 #10-208                    |
| TRANS      | #5-57 #9-158 #9-160 #9-162 #9-164 #9-166 #9-168 #9-170 #9-171 #9-174  |
|            | #9-175 #9-178 #9-179 #9-182 #9-183 #9-186 #9-187 #9-190 #9-191 #9-194 |
|            | #9-195 #9-198 #10-205 #10-206   |

EVL            CREATED BY MACRO ON 28-JUN-85 AT 22:58      PAGE 7      F 1  
SYMBOL CROSS REFERENCE                                  CREF 04.00

| SYMBOL  | VALUE       | REFERENCES     |
|---------|-------------|----------------|
| Z.NAM   | 000004      | #5-62          |
| Z.PCB   | 000012      | #5-62          |
| Z.SCH   | 000007      | #5-62          |
| SEACC   | = *****     | 8-95           |
| SEVLTB  | = 000000 GX | #7-77          |
| SEVLVL  | = 000013 RG | #8-114         |
| SEVI VT | = 000012 G  | #8-85 8-114    |
| SEXRON  | = ***** GX  | 8-96           |
| SLGDDB  | = ***** GX  | 8-98           |
| SLGST   | = ***** GX  | 11-350 *11-356 |
| SLGPDV  | = ***** GX  | 8-99           |
| SLGSTT  | = ***** GX  | 8-100          |
| LOST    | = ***** GX  | 11-335 11-352  |
| SPDVID  | = ***** GX  | 8-108          |
| SPDVTA  | = ***** GX  | 8-109          |
| SPSIPT  | = ***** GX  | 8-110          |
| SLTMA   | = ***** GX  | 8-111          |
| STNS    | = ***** GX  | 8-94           |
| SZTIM2  | = ***** GX  | 8-112          |

EVL                      CREATED BY   MACRO   ON 28-JUN-85 AT 22:58                      PAGE 8                      G 1

ALCM  
Sym

|                |                  |                 |                |                 |
|----------------|------------------|-----------------|----------------|-----------------|
| ASSCHK= 000000 | G\$TSS= 000000   | L\$11R= 000000  | PIRQ = 177772  | R\$SND= 000000  |
| ASSCPS= 000000 | G\$TTK= 000000   | MPAR = 172100   | Pmode = 030000 | R\$11M= 000000  |
| ASSPRI= 000000 | G\$WRD= 000000   | MPCSR = 177746  | PR0 = 000000   | SWR = 177570    |
| ASSTRP= 000000 | I\$RAR= 000000   | M\$CRB= 000124  | PR1 = 000040   | S\$WRG= 000000  |
| BUFJMP= 172354 | I\$RDN= 000000   | M\$CRX= 000000  | PR2 = 000100   | S\$YSZ= 007600  |
| Cmode = 140000 | KISAR0= 172340   | M\$FCFS= 000000 | PR3 = 000140   | TPS = 177564    |
| C\$CKP= 000000 | KISAR6= 172354   | M\$MGE= 000000  | PR4 = 000200   | T\$KMG= 000000  |
| C\$ORE= 000400 | KSAR6 = ***** GX | M\$NET= 000000  | PR5 = 000240   | T\$MIN= 000000  |
| C\$RSH= 177564 | KS6 = 000000R    | M\$GVR= 000000  | PR6 = 000300   | UBMPR = 170200  |
| D\$BUG= 177514 | K\$CNT= 177546   | N\$ACC= 000001  | PR7 = 000340   | UISAR0= 177640  |
| D\$ISK= 000000 | K\$CSR= 177546   | N\$BUF= 000001  | PS = 177776    | UISAR1= 177642  |
| D\$LI1= 000001 | K\$LDC= 000000   | N\$LDV= 000001  | P\$P45= 000000 | V\$CTR= 001000  |
| D\$YNC= 000000 | K\$TPS= 000074   | N\$MCP= 000001  | P\$WRD= 000000 | X\$DBT= 000000  |
| D\$YNM= 000000 | LD\$LP = 000000  | N\$MLL= 000001  | Q\$OPT= 000010 | \$AMEM 000002RG |
| E\$XPR= 000000 | L\$ASG= 000000   | N\$MOV= 000010  | R\$DER= 000000 | \$DMEM 000220RG |
| F\$LVL= 000001 | L\$DRV= 000000   | N\$NCT= 000001  | R\$K11= 000001 | .BASEB= 140000  |
| G\$TPP= 000000 | L\$P11= 000001   | N\$PEM= 000001  |                |                 |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
 000364 001 (RW,I,LCL,REL,CON)

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 9770 Words ( 39 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:09.50

SY:ALCMEM.V2,[132,134]ALCMEM/CR/-SP=SY:[1,1]RSXCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXCM/PA:1,[132,10]ALCMEM

CFGENT - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 <sup>F 3</sup> 00:01  
Table of contents

|    |     |                                 |
|----|-----|---------------------------------|
| 5- | 53  | MACRO DEFINITIONS               |
| 6- | 70  | ERROR MESSAGES                  |
| 7- | 91  | \$OCNT - LOOK FOR CNT\$DF MACRO |
| 8- | 121 | TPARS STATE TABLES              |
| 9- | 166 | CNT\$DF ACTION ROUTINES         |

CFGDDM - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 <sup>F 4</sup> 00:02  
Table of contents

|    |     |                                 |
|----|-----|---------------------------------|
| 5- | 53  | MACRO DEFINITIONS               |
| 6- | 60  | \$QDDM - LOOK FOR DDM\$DF MACRO |
| 7- | 90  | TPARS STATE TABLES              |
| 8- | 152 | ACTION ROUTINES                 |



```

54      .SBTTL  MACRO DEFINITIONS
55
56      ;
57      ; LIBRARY MACROS
58      ;
59      .MCALL  MSG$R,SLTDF$,SAVRG,RESRG,DTEDF$,RETC
60      .MCALL  PHBDF$,NTLR$,ISTAT$,TRAN$,STAT$,ASL$
61
62      ; LIBRARY SYMBOLS
63      ;
64      DTEDF$      ; LOCAL DTE DESCRIPTOR BLOCK OFFSETS
65      PHBDF$      ; DEFINE PSI HOME BLOCK OFFSETS
66      SLTDF$      ; DEFINE SLT OFFSETS
67
68      ; LOCAL MACRO DEFINITIONS
69      ;
70      .MACRO  SAVMAP
71      MOV     @KSAR5,-(SP)      ; SAVE APR 5
72      .ENDM
73
74      .MACRO  RESMAP
75      MOV     (SP)+,@KSAR5      ; RESTORE APR5
76      .ENDM

```

```

416 .SBTTL LINE-ID ACTION ROUTINES
417 .ENABL LSB
418 ;
419 ; DEVICE NAME
420 ;
421 000520 022767 000003 000000G DEVNAM: CMP #3, PSTCN ; VALID DEVICE NAME?
422 000526 103425 BLO 101$ ; BR IF NO
423 000530 016700 000000G MOV , PSTPT, R0 ; GET ADDRESS OF FIRST CHARACTER
424 000534 CALL $CAT5 ; CONVERT TO RAD50
425 000540 010167 000042' MOV R1, LINNAM ; SAVE DEVICENAME
426 000544 RETURN
427 ;
428 ; DEVICE CONTROLLER NUMBER
429 ;
430 000546 005767 000000G DEVCTL: TST , PNUMH ; LEGAL CONTROLLER NUMBER VALUE?
431 000552 001013 BNE 101$ ; BR IF NO
432 000554 016767 000000G 000044' MOV , PNUMB, LINCTL ; SAVE CONTROLLER NUMBER
433 000562 RETURN
434 ;
435 ; DEVICE UNIT NUMBER
436 ;
437 000564 005767 000000G DEVUNT: TST , PNUMH ; LEGAL UNIT NUMBER VALUE?
438 000570 001004 BNE 101$ ; BR IF NO
439 000572 116767 000000G 000045' MOVB , PNUMB, LINUNT ; SAVE UNIT NUMBER
440 000600 RETURN
441 ;
442 ; ERRORS
443 ;
444 000602 101$: MSG$R 06 ; ILLEGAL LINE-ID
445 .DSABL LSB

```

```

73                                     .SBTTL  $QFEA - LOOK FOR FEASDF MACRO
74
75                                     :+
76                                     : $QFEA - LOOK FOR FEASDF MACRO
77                                     :
78                                     : INPUTS:
79                                     :   NONE
80                                     :
81                                     : OUTPUTS:
82                                     :   ALL REGISTERS DESTROYED
83                                     :
84                                     :-
85
86 000000 012705 000000' $QFEA:: MOV    #FEADF,R5    ; STATE TABLE ADDRESS
87 000004 005001          CLR    R1              ; FULL KEYWORD MATCH LENGTH
88 000006 012702 000000' MOV    #FEAKW,R2      ; KEYWORD TABLE ADDRESS
89 000012 016703 000000G MOV    CFGSZ,R3      ; RECORD LENGTH
90 000016 012704 000000G MOV    #CFGBF,R4      ; RECORD BUFFER ADDRESS
91 000022 005067 000000G CLR    SYNERR        ; CLEAR SYNTAX ERROR FLAG
92 000026          CALL  TPARS          ; GO DO THE PARSE
93 000032 103003          BCC    20$         ; IF CC, FOUND WHAT WE WERE LOOKING FOR
94 000034 005367 000000G DEC    SYNERR        ; DID SYNTAX ERROR OCCUR?
95 000040 001401          BEQ    101$        ; IF EQ, YES
96 000042          20$:  RETURN
97
98                                     : ERRORS
99
100 000044          101$:  MSG$R 1T          ; SYNTAX ERROR

```

```

174 .SBTTL LOOK FOR PVC$DF MACRO
175
176 ;+
177 $QPVC - LOOK FOR PVC$DF MACRO
178
179 ; INPUTS:
180 ; NONE
181
182 ; OUTPUTS:
183 ; C-BIT=SUCCESS/FAILURE
184 ; R3,R4,R5=DESTROYED
185
186 000000 005067 000032' $QPVC:: CLR FLAG ; CLEAR LOCAL FLAGS WORD
187 000004 012705 000000' MOV #PVCDF,R5 ; STATE TABLE ADDRESS
188 000010 005001 MOV R1 ; FULL KEYWORD MATCH LENGTH
189 000012 012702 000000' MOV #PCVKW,R2 ; KEYWORD TABLE ADDRESS
190 000016 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
191 000022 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
192 000026 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
193 000032 CALL TPARS ; GO DO THE PARSE
194 000036 103003 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
195 000040 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
196 000044 001401 BEQ 101$ ; IF EQ, YES
197 000046
198
199 ; ERROR CONDITION
200
201 101$: MSG$R 17 ; SYNTAX ERROR
202 000050

```

```

698 .SBTTL PVDLM - CHECK FOR DLM PVC
699
700
701
702 PVDLM - CHECK FOR DLM PVC
703
704 INPUTS:
705 PVCNAM - PROCESS NAME (ASCII)
706
707 OUTPUTS:
708 DLMPDV - DLM'S PDV ADDRESS
709
710
711 PVDLM: MOV #PVCNAM,R0 ; POINT TO SPECIFIED PROCESS NAME
712 CALL $CAT5 ; CONVERT TO RAD50
713 CMP R1,#^RDLM ; IS THIS DLM?
714 BNE INVCIR ; BR IF NO - IGNORE IT
715 MOV R1,R2 ; GET DLM'S RAD50 PROCESS NAME
716
717 SWSTK$ 20$ ; ENTER SYSTEM STATE
718 CALL @PDVID ; GET DLM'S PDV INDEX
719 BCC 10$ ; BR IF FOUND IT
720 RETC R0 ; ELSE SET USER C-BIT
721 BR 15$ ; AND EXIT
722 MOV R2,DLMPDV ; SAVE PDV INDEX
723 ADD @PDVTA,R2 ; POINT TO ENTRY IN PDV VECTOR TABLE
724 MOV (R2),DLMPDV ; SAVE DLM'S PDV INDEX
725 RETURN ; EXIT SYSTEM STATE
726
727 20$: BCS INVCIR ; BR IF DLM NOT IN SYSTEM
728 RETURN
729
730 ERRORS
731
732 INVCIR: MSG$R NO ; ILLEGAL CIRCUIT ID
733
734 PVCTL - GET CONTROLLER NUMBER FOR DLM
735
736 .ENABL LSB
737 PVCTL: MOVB .PNUMB,DLMCTL ; SAVE CONTROLLER NUMBER
738 TST .PNUMH ; LEGAL VALUE?
739 BNE INVCIR ; BR IF NO - ERROR
740 TSTB .PNUMB+1 ; LEGAL VALUE?
741 BNE INVCIR ; BR IF NO
742 MOV $SLTA,R0 ; GET SLT ADDRESS
743 CMPB .PNUMB,L.CTL(R0) ; IS THIS THE CORRECT CONTROLLER NUMBER?
744 BEQ 7$ ; CHECK FOR VALIDITY
745 CLR SYNERR ; ELSE NO SYNTAX ERROR
746 ADD #2,(SP) ; AND REJECT TRANSITION
747 BR 20$ ; ...
748 RETURN
749 PVSTA: MOVB .PNUMB,DLMSTA ; SAVE STATION NUMBER
750 CMPB .PNUMB,$TRIB ; CORRECT STATION NUMBER?
751 BNE 5$ ; BR IF NO
752 TST .PNUMH ; LEGAL VALUE?
753 BNE INVCIR ; BR IF NO - ERROR
754 TSTB .PNUMB+1 ; LEGAL VALUE?

```

CFGP  
MACR  
MACR  
BIAS  
CALL

CTRD  
DBGT

DTED  
EMSG

ISTA  
MAP  
MTRA  
NTLE  
PHBC  
PLBU  
PVED  
RESM  
RESR  
RETC  
RETL

SAVM  
SAVR  
SLTC  
SOB  
STAT

SWST  
TRAN

XCBI  
XPDI

CFGPVC CREATED BY MACRO ON 29-JUN-85 AT 00:04 PAGE 7 F 10  
 MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES  |
|------------|---|
| BIAS       | #5-92 21-807  |
| CALL       | 9-193 12-353 12-365 12-381 13-413 13-417 13-421 13-427 13-439 13-445            |
|            | 13-448 13-455 14-482 14-486 14-489 14-490 14-498 15-525 15-530 16-569           |
|            | 17-603 17-611 18-648 20-712 20-717 20-718                                       |
| CTRDF\$    | #5-61 5-66  |
| DBGTP\$    | #10-208 #10-236 #10-240 #10-244 #10-248 #10-252 #10-256 #10-260 #10-264 #10-268 |
|            | #10-272   |
| DIEDF\$    | #5-61 5-65  |
| EMSG\$R    | #5-59 9-202 11-319 11-335 11-347 12-360 12-372 12-387 12-401 13-464             |
|            | 13-465 20-732   |
| ISTAT\$    | #5-60 10-208  |
| MAP        | #5-86 21-804  |
| MTRANS     | #10-208   |
| NILERS     | #5-60 8-161 8-162 8-163 8-164 8-165 8-166 8-167 8-168                           |
| PHBDF\$    | #5-60 5-68  |
| PLBDF\$    | #5-61 5-72  |
| PVCDF\$    | #5-59 5-69  |
| RESMAP     | #5-80 13-456 15-545 16-583 18-675 21-820  |
| RESRG      | #5-59   |
| RETC       | #5-59 13-423 13-441 13-450 15-544 20-720  |
| RETURN     | 9-197 11-315 11-331 11-343 12-356 12-368 12-383 12-397 13-457 13-460            |
|            | 14-505 15-546 16-584 17-628 18-676 19-696 20-725 20-728 20-748 20-756           |
|            | 21-821  |
| SAVMAP     | #5-76 13-418 15-526 16-565 18-646 21-802  |
| SAVRG      | #5-59   |
| SLTDF\$    | #5-61 5-70  |
| SOB        | 21-797 21-813   |
| STATE\$    | #5-60 10-213 #10-216 #10-219 #10-222 #10-225 #10-228 #10-231 #10-234 #10-238    |
|            | #10-242 #10-246 #10-250 #10-254 #10-258 #10-262 #10-266 #10-270 #10-276 #10-279 |
|            | #10-282 #10-285 #10-288 #10-291   |
|            | 13-417 15-525 20-717  |
| SWSTK\$    | #5-60 #10-214 #10-217 #10-220 #10-223 #10-226 #10-229 #10-232 #10-235 #10-236   |
| TRANS      | #10-239 #10-240 #10-243 #10-244 #10-247 #10-248 #10-251 #10-252 #10-255 #10-256 |
|            | #10-259 #10-260 #10-263 #10-264 #10-267 #10-268 #10-271 #10-272 #10-277 #10-280 |
|            | #10-283 #10-286 #10-289   |
| XCBDF\$    | #5-59 5-71  |
| XPDD\$     | #5-61 5-67  |

CFGSLT      CREATED BY    MACRO    ON 29-JUN-85 AT 00:05      PAGE 3      F 11

MACRO CROSS REFERENCE      CREF    04.00

MACRO NAME      REFERENCES

|         |        |        |        |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| CALL    | 6-92   |        |        |        |        |        |        |        |        |        |
| DBGTP\$ | #7-107 | #7-136 | #7-139 | #7-142 | #7-145 | #7-148 | #7-151 | #7-154 | #8-170 | #8-171 |
|         | #8-174 | #8-175 | #8-178 | #8-184 |        |        |        |        |        |        |
| EMSG\$R | #5-56  | 6-101  | 9-211  | 9-223  |        |        |        |        |        |        |
| ISTAT\$ | #5-56  | 7-107  |        |        |        |        |        |        |        |        |
| MTRANS  | #7-107 |        |        |        |        |        |        |        |        |        |
| REJ\$   | #5-66  | 9-194  | 9-202  | 9-241  |        |        |        |        |        |        |
| RETURN  | 6-96   | 9-195  | 9-203  | 9-210  | 9-221  | 9-231  | 9-239  | 9-242  |        |        |
| SLTDF\$ | #5-56  | 5-58   |        |        |        |        |        |        |        |        |
| STAT\$  | #5-56  | 7-112  | #7-114 | #7-116 | #7-118 | #7-120 | #7-122 | #7-124 | #7-126 | #7-128 |
|         | #7-130 | #7-132 | #7-134 | #7-137 | #7-140 | #7-143 | #7-146 | #7-149 | #7-152 | #7-155 |
|         | #7-157 | #7-159 | 8-168  | #8-172 | #8-176 | #8-182 | #8-186 |        |        |        |
| TRANS   | #5-56  | #7-113 | #7-115 | #7-117 | #7-119 | #7-121 | #7-123 | #7-125 | #7-127 | #7-129 |
|         | #7-131 | #7-133 | #7-135 | #7-136 | #7-138 | #7-139 | #7-141 | #7-142 | #7-144 | #7-145 |
|         | #7-147 | #7-148 | #7-150 | #7-151 | #7-153 | #7-154 | #7-156 | #7-158 | #7-160 | #8-169 |
|         | #8-170 | #8-171 | #8-173 | #8-174 | #8-175 | #8-177 | #8-178 | #8-183 | #8-184 |        |

CFGSTA      CREATED BY MACRO ON 29-JUN-85 AT 00:06      PAGE 1      F 12

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL | VALUE      | REFERENCES                                   |
|--------|------------|--|
| CERR   | = ***** GX | 7-75      7-76      7-77      7-78      7-79 |
| CFGBF  | = ***** GX | 8-100  |
| CFGSZ  | = ***** GX | 8-99   |
| CFLIN  | = ***** GX | 7-75      7-76      7-77      7-78      7-79 |
| FMT8   | = ***** GX | 7-76   |
| FMT8B  | = ***** GX | 7-75      7-77      7-78      7-79           |
| FM.8   | = 000000   | #7-76  |
| FM.8B  | = 000000   | #7-75      #7-77      #7-78      #7-79       |
| LF.ACT | = 100000   | #5-63  |
| LF.BRO | = 000400   | #5-63  |
| LF.BWT | = 000007   | #5-63  |
| LF.ENA | = 002000   | #5-63  |
| LF.LPB | = 001000   | #5-63  |
| LF.MDC | = 000100   | #5-63  |
| LF.MFL | = 004000   | #5-63  |
| LF.MTP | = 000020   | #5-63      10-201                            |
| LF.PAC | = 000200   | #5-63  |
| LF.RDY | = 040000   | #5-63  |
| LF.REA | = 010000   | #5-63  |
| LF.SER | = 000040   | #5-63  |
| LF.TIM | = 000010   | #5-63  |
| LF.UNL | = 020000   | #5-63  |
| LF.X2P | = 000000   | #5-63  |
| LN.CLO | = 000000   | #5-63  |
| LN.DUM | = 0000C5   | #5-63  |
| LN.LOA | = 000004   | #5-63  |
| LN.LOO | = 000003   | #5-63  |
| LN.OAU | = 000003   | #5-63  |
| LN.OFF | = 000001   | #5-63  |
| LN.ON  | = 000000   | #5-63  |
| LN.OOP | = 000004   | #5-63  |
| LN.OPE | = 000001   | #5-63  |
| LN.REF | = 000002   | #5-63  |
| LN.SER | = 000002   | #5-63  |
| LN.STA | = 000017   | #5-63  |
| LN.SUB | = 000360   | #5-63  |
| LN.TRI | = 000006   | #5-63  |
| L.COST | 000015     | #5-63  |
| L.CTL  | 000012     | #5-63  |
| L.CVA  | 177776     | #5-63  |
| L.DDM  | 000002     | #5-63  |
| L.DDS  | 000004     | #5-63  |
| L.DLC  | 000003     | #5-63  |
| L.DLM  | 000006     | #5-63  |
| L.DLS  | 000010     | #5-63  |
| L.FLG  | 000000     | #5-63      10-201                            |
| L.KRBA | 000016     | #5-63  |
| L.LEN  | = 000022   | #5-63  |
| L.MPF  | 000022     | #5-63      10-210                            |
| L.NMST | 000020     | #5-63  |
| L.NSTA | 000014     | #5-63      10-203                            |
| L.OWNR | 000021     | #5-63  |



```

255          .SBTTL  SVC$DF ACTION ROUTINES
256
257          .ENABL  LSB
258
259          ; SVNAM - PROCESS NAME
260
261 000052 022767 000003 000000G SVNAM: CMP    #3, PSTCN      ; VALID NAME?
262 000060 001063          BNE    101$          ; BR IF NO
263 000062 016700 000000G          MOV    , PSTPT, R0    ; POINT TO PROCESS NAME
264 000066 005001          CLR    R1
265 000070          CALL    $CAT5          ; CONVERT TO RAD50
266 000074 022701 015355          CMP    #^RDLM, R1    ; IS IT DLM?
267 000100 001053          BNE    101$          ; BR IF NO - ERROR
268 000102 005067 000006'          CLR    PKTSVC        ; INITIALIZE OPTIONAL PARAMETERS
269 000106 105067 000022'          CLRB   WND$VC        ; ...
270 000112          RETURN
271
272          ; SVCTL - CONTROLLER NUMBER
273
274
275 000114 005767 000000G SVCTL: TST    , PNUMH          ; VALID LINE-ID?
276 000120 001043          BNE    101$          ; BR IF NO
277 000122 105767 000001G          TSTB   , PNUMB+1      ; ...
278 000126 001040          BNE    101$          ; BR IF INVALID LINE-ID
279 000130 116767 000000G 000021'        MOVB   , PNUMB, CTLSVC    ; SAVE CONTROLLER NUMBER
280 000136 016700 000000G          MOV    $SLTA, R0    ; GET SLT ADDRESS
281 000142 126760 000000G 000012        CMPB   , PNUMB, L.CTL(R0) ; CORRECT CONTROLLER NUMBER?
282 000150 001404          BEQ     10$           ; BR IF YES
283 000152          REJ$          ; ELSE REJECT TRANSITION
284 000162          10$: RETURN
285
286          ; SVSTA - STATION NUMBER
287
288
289 000164 005767 000000G SVSTA: TST    , PNUMH          ; VALID LINE-ID?
290 000170 001017          BNE    101$          ; BR IF NO
291 000172 105767 000001G          TSTB   , PNUMB+1      ; ...
292 000176 001014          BNE    101$          ; BR IF INVALID ID
293 000200 116767 000000G 000014'        MOVB   , PNUMB, STASVC    ; SAVE STATION NUMBER
294 000206 126767 000000G 000000G        CMPB   $TRIB, PNUMB    ; CORRECT STATION NUMBER?
295 000214 001404          BEQ     20$           ; BR IF YES
296 000216          REJ$          ; ELSE REJECT TRANSITION
297 000226          20$: RETURN
298
299          ; ERROR
300
301 000230          101$: EM$G$R  2I          ; ILLEGAL LINE-ID
302          .DSABL  LSB

```

CFGSVC CREATED BY MACRO ON 29-JUN-85 AT 00:07 PAGE 2 F 14  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                      |
|---------|------------|---|
| LF.UNL  | = 020000   | #5-65   |
| LF.X2P  | = 000000   | #5-65   |
| LLCTA   | = ***** GX | 15-542  |
| LN.CLO  | = 000000   | #5-65   |
| LN.DUM  | = 000005   | #5-65   |
| LN.LOA  | = 000004   | #5-65   |
| LN.LOO  | = 000003   | #5-65   |
| LN.OAU  | = 000003   | #5-65   |
| LN.OFF  | = 000001   | #5-65   |
| LN.ON   | = 000000   | #5-65   |
| LN.OOP  | = 000004   | #5-65   |
| LN.OPE  | = 000001   | #5-65   |
| LN.REF  | = 000002   | #5-65   |
| LN.SER  | = 000002   | #5-65   |
| LN.STA  | = 000017   | #5-65   |
| LN.SUB  | = 000360   | #5-65   |
| LN.TRI  | = 000006   | #5-65   |
| L.TMR   | = 000004 R | #6-117 *13-430 15-581                           |
| L.COST  | = 000015   | #5-65   |
| L.CTL   | = 000012   | #5-65 10-281                                    |
| L.CVA   | = 177776   | #5-65   |
| L.DDM   | = 000002   | #5-65   |
| L.DDS   | = 000004   | #5-65   |
| L.DLC   | = 000003   | #5-65   |
| L.DLM   | = 000006   | #5-65   |
| L.DLS   | = 000010   | #5-65   |
| L.FLG   | = 000000   | #5-65   |
| L.KRBA  | = 000016   | #5-65   |
| L.LEN   | = 000022   | #5-65   |
| L.MPF   | = 000022   | #5-65   |
| L.NMST  | = 000020   | #5-65   |
| L.NSTA  | = 000014   | #5-65   |
| L.OWNR  | = 000021   | #5-65   |
| L.UNT   | = 000013   | #5-65   |
| MAXWND  | = 000000   | #6-108 11-322                                   |
| NEXT    | = 000000 R | #6-115 *12-388 12-395 17-640 *17-641            |
| NODTE   | = 000420 R | #12-372   |
| NSTLC   | = 000100   | 15-561 15-562                                   |
| NSSVCT  | = *****    | 13-439 14-515                                   |
| OWNNSVC | = 000020 R | #6-123 *14-516 15-553                           |
| PCKBCD  | = 001440 R | 12-411 #16-601                                  |
| PDVID   | = ***** GX | 14-513  |
| PDVTA   | = ***** GX | 15-554  |
| PKTSVC  | = 000006 R | #6-118 *10-268 *11-310 13-455                   |
| PSIPT   | = ***** GX | 13-431  |
| PXSDLM  | = 000200   | 15-572  |
| PXSSVC  | = 000100   | 15-572  |
| RCL SVC | = 000023 R | #6-126 *11-350 13-457                           |
| RTSPC   | = ***** GX | 7-138 7-140 7-141 7-142 7-143 7-144 7-145 7-146 |
| RTYSVC  | = 000012 R | #6-120 *12-362 13-459                           |
| RSS EIS | = *****    | 16-612  |

CFGUNF - CONFIG FILE SCAN ACTIO MACRO V05.03b Monday 15-Jul-85 18:56 Page 10  
UNF\$DF STATE TABLE

```
201  
202  
203  
204 000112  
205 000112  
206 000112  
207  
208 000112  
:  
: CHECK FOR END OF SOURCE LINE  
:  
STATE$ END  
TRANS <'.>,$EXIT  
TRANS $EOS,$EXIT  
STATE$
```

\*\*F

C  
C  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
C  
C

LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL  
LL

\*\*FILE\*\*ID\*\*CFGX2P

```

CCCCCCCC FFFFFFFF GGGGGGGG XX XX 222222 PPPPPPPP
CCCCCCCC FFFFFFFF GGGGGGGG XX XX 222222 PPPPPPPP
CC FF GG XX XX 22 22 PP PP
CC FF GG XX XX 22 22 PP PP
CC FF GG XX XX 22 22 PP PP
CC FF GG XX XX 22 22 PP PP
CC FFFFFFFF GG XX XX 22 22 PPPPPPPP
CC FFFFFFFF GG XX XX 22 22 PPPPPPPP
CC FF GG GGGGGG XX XX 22 22 PP
CC FF GG GGGGGG XX XX 22 22 PP
CC FF GG GG XX XX 22 22 PP
CC FF GG GG XX XX 22 22 PP
CC CCCCCC FF GGGGGG XX XX 2222222222 PP
CCCCCCCC FF GGGGGG XX XX 2222222222 PP

```

```

....
....
....
....

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

CREF 04.00

| MACRO NAME | REFERENCES |
|------------|------------|
|            |            |

| CALL    | 9-146<br>13-433 | 9-179<br>13-448 | 9-183  | 9-211  | 9-239  | 10-295 | 10-313 | 11-353 | 11-375 | 13-427 |
|---------|-----------------|-----------------|--------|--------|--------|--------|--------|--------|--------|--------|
| CCBDF\$ | #5-51           | 5-63            |        |        |        |        |        |        |        |        |
| DLMD\$  | #5-54           | 5-59            |        |        |        |        |        |        |        |        |
| ENABL\$ | #5-51           | 9-223           | 11-368 |        |        |        |        |        |        |        |
| ENDEVE  | #6-71           | 9-276           |        |        |        |        |        |        |        |        |
| EVENT   | #6-67           | 9-275           |        |        |        |        |        |        |        |        |
| EVLD\$  | #5-51           | 5-61            |        |        |        |        |        |        |        |        |
| INHIB\$ | #5-51           | 9-199           | 11-338 |        |        |        |        |        |        |        |
| MAP     | #5-51           | 9-174           | 13-462 |        |        |        |        |        |        |        |
| PDVDF\$ | #5-51           | 5-62            |        |        |        |        |        |        |        |        |
| PHBDF\$ | #5-54           | 5-56            |        |        |        |        |        |        |        |        |
| PVCD\$  | #5-54           | 5-58            |        |        |        |        |        |        |        |        |
| RESMAP  | #5-52           | 13-473          |        |        |        |        |        |        |        |        |
| RSRG    | #5-51           | 9-243           | 9-264  | 11-376 | 12-405 | 14-504 |        |        |        |        |
| RETURN  | 9-269           | 10-315          | 11-379 | 12-406 | 13-474 | 14-505 |        |        |        |        |
| SAVMAP  | #5-52           | 13-421          |        |        |        |        |        |        |        |        |
| SAVRG   | #5-51           | 9-149           | 9-261  | 11-373 | 12-395 | 14-491 |        |        |        |        |
| SLTDF\$ | #5-51           | 5-60            |        |        |        |        |        |        |        |        |
| SOB     | 9-197           | 9-263           |        |        |        |        |        |        |        |        |
| XCBDF\$ | #5-54           | 5-57            |        |        |        |        |        |        |        |        |

| SYMBOL | VALUE      | REFERENCES |        |        |        |        |       |        |       |       |
|--------|------------|------------|--------|--------|--------|--------|-------|--------|-------|-------|
| BUFUMP | = 172354   | #5-117     |        |        |        |        |       |        |       |       |
| CMODE  | = 140000   | #5-117     |        |        |        |        |       |        |       |       |
| IS\$AS | = *****    | 5-117      |        |        |        |        |       |        |       |       |
| KISARO | = 172340   | #5-117     |        |        |        |        |       |        |       |       |
| KISAR6 | = 172354   | #5-117     |        |        |        |        |       |        |       |       |
| KSAR6  | = ***** GX | 6-160      | 7-258  |        |        |        |       |        |       |       |
| KS6    | 000000 R   | #5-119     | *6-160 | 6-161  | 6-166  | 6-179  | 6-181 | 6-187  | 6-193 | 6-194 |
|        |            | 6-206      | 6-208  | 6-211  | 6-215  | 6-217  | 6-225 | *7-258 | 7-259 | 7-262 |
|        |            | 7-365      |        |        |        |        |       |        |       |       |
| MPAR   | = 172100   | #5-117     |        |        |        |        |       |        |       |       |
| MPCSR  | = 177746   | #5-117     |        |        |        |        |       |        |       |       |
| M\$MGE | = 000000   | 5-117      |        |        |        |        |       |        |       |       |
| PIRQ   | = 177772   | #5-117     |        |        |        |        |       |        |       |       |
| PMODE  | = 030000   | #5-117     |        |        |        |        |       |        |       |       |
| PRO    | = 000000   | #5-117     |        |        |        |        |       |        |       |       |
| PR1    | = 000040   | #5-117     |        |        |        |        |       |        |       |       |
| PR2    | = 000100   | #5-117     |        |        |        |        |       |        |       |       |
| PR3    | = 000140   | #5-117     |        |        |        |        |       |        |       |       |
| PR4    | = 000200   | #5-117     |        |        |        |        |       |        |       |       |
| PR5    | = 000240   | #5-117     |        |        |        |        |       |        |       |       |
| PR6    | = 000300   | #5-117     |        |        |        |        |       |        |       |       |
| PR7    | = 000340   | #5-117     |        |        |        |        |       |        |       |       |
| PS     | = 177776   | #5-117     |        |        |        |        |       |        |       |       |
| R\$MPL | = *****    | 5-117      | 5-117  |        |        |        |       |        |       |       |
| R\$11D | = *****    | 5-117      |        |        |        |        |       |        |       |       |
| SWR    | = 177570   | #5-117     |        |        |        |        |       |        |       |       |
| TPS    | = 177564   | #5-117     |        |        |        |        |       |        |       |       |
| UBMPR  | = 170200   | #5-117     |        |        |        |        |       |        |       |       |
| UISARO | = 177640   | #5-117     |        |        |        |        |       |        |       |       |
| UISAR1 | = 177642   | #5-117     |        |        |        |        |       |        |       |       |
| \$AMEM | 000002 RG  | #6-158     |        |        |        |        |       |        |       |       |
| \$DMEM | 000220 RG  | #7-257     |        |        |        |        |       |        |       |       |
| .BASEB | = 140000   | #5-117     | 6-171  | 6-180  | *6-182 | *6-183 | 6-188 | 6-194  | 6-198 | 6-207 |
|        |            | *6-209     | *6-210 | *6-212 | 6-216  | *6-218 | 7-264 |        |       |       |

.TITLE CFGCNT - SCAN CONFIGURATION FILE FOR DYNAMIC INFORMATION  
.IDENT /V05.00/

COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE SCAN FOR "SET LINE" AND "SET PROCESS"

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 06-AUG-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

.TITLE CFGDDM - SCAN CONFIGURATION FILE FOR DYNAMIC INFORMATION  
.IDENT /V05.00/

COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE SCAN FOR "SET LINE" AND "SET PROCESS"

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 06-AUG-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RX V1.0



```

LOCAL SYMBOL DEFINITIONS

78          .SBTTL LOCAL SYMBOL DEFINITIONS
79
80          ;
81          ; LOCAL SYMBOL DEFINITIONS
82          ;
83          000040      SPACE = 40          ; ASCII SPACE
84          000040      HSHMN = 32          ; MINIMUM VALUE FOR HASH TABLE SIZE
85          001000      HSHMX = 512        ; MAXIMUM VALUE FOR HASH TABLE SIZE
86          000020      NUMDTE = 16        ; MAXIMUM DTES ALLOWED
87
88          ;
89          ; SAVED REGISTER OFFSETS ON STACK FOR SWSTK$
90          ;
91          000002      R$R0 = 2            ; SAVED R0
92          000004      R$R1 = 4            ; SAVED R1
93          000006      R$R2 = 6            ; SAVED R2
94          000010      R$R3 = 10           ; SAVED R3
95          000012      R$R4 = 12           ; SAVED R4
96          000014      R$R5 = 14           ; SAVED R5

```

```

447                                     .SBITL  STRNXT - STORE NEXT DIGIT
448
449                                     ;+
450                                     ; STRNXT - STORE NEXT DIGIT
451                                     ;
452                                     ; INPUTS:
453                                     ; .PCHAR - MATCHED CHARACTER FROM TPARS TRANSITION
454                                     ; NEXT - ADDRESS OF NEXT BYTE AVAILABLE FOR STORAGE
455                                     ;
456                                     ; OUTPUTS:
457                                     ; NEXT - ADDRESS OF NEXT AVAILABLE BYTE
458                                     ; COUNT - UPDATED
459                                     ; RO,R1 DESTROYED
460                                     ;
461                                     ;-
462 000610 116767 000000G 000036' STRNXT: MOVB    .PCHAR,CVTBUF      ; MOVE DIGIT TO CONVERSION BUFFER
463 000616 012701 000017      MOV     #17,R1          ; ASSUME CHAR IS AN '*'
464 000622 126727 000000G 000052      CMPB    .PCHAR,#'*      ; IS CHAR AN '*' ?
465 000630 001404      BEQ     10$                    ; BR IF YES
466 000632 012700 000036'      MOV     #CVTBUF,R0        ; POINT TO CONVERSION BUFFER
467 000636      CALL    $CDTB                          ; CONVERT TO BINARY
468 000642 110177 000040'      MOVB    R1,@NEXT          ; STORE BINARY DIGIT
469 000646 005267 000040'      INC      NEXT            ; POINT TO NEXT AVAILABLE BYTE
470 000652 105267 000035'      INCB    COUNT            ; INCREMENT DIGIT COUNT
471 000656      RETURN

```

```

102          .SETTL FEASDF STATE TABLE
103          :
104          : TPARS STATE TABLES
105          :
106          : ISTAT$ FEAST,FEAKW
107          :
108          :
109          : FEATURES DEFINITION (FEASDF)
110          :
111          : STATES$ FEADF
112          : TRANS$ %FEASDF%,...1,SYNERR
113          :
114          : STATES$
115          : TRANS$ $ALPHA,,F.CH1
116          :
117          : STATES$
118          : TRANS$ $ALPHA,,F.CH2
119          :
120          : STATES$
121          : TRANS$ <','>
122          :
123          : STATES$
124          : TRANS$ $NUMBR,,F.FEA
125          :
126          : STATES$
127          : TRANS$ <','>
128          :
129          : STATES$
130          : TRANS$ $NUMBR,,F.MOD
131          :
132          : STATES$
133          : TRANS$ <','>
134          :
135          : STATES$
136          : TRANS$ $NUMBR,$EXIT,F.SET
137          :
138          : STATES$

```

```

204 .SBTTL TPARS STATE TABLES
205 ;
206 ; TPARS STATE TABLES
207 ;
208 000056 ISTAT$ FVCST,PCVKW
209 ;
210 ; PVC$DF STATE TABLES
211 ;
212 ;
213 000056 STATES$ PVCDF
214 000056 TRANS$ %PVC$DF%
215 ;
216 000056 STATES$
217 000056 TRANS$ $RAD50,,PVNAM ; CHANNEL IDENTIFICATION
218 ;
219 000056 STATES$
220 000056 TRANS$ !CHKDLM,CHNLNO
221 ;
222 000056 STATES$ CHNLNO
223 000056 TRANS$ $NUMBR,,PVLN,1,SYNERR ; LOGICAL CHANNEL NUMBER
224 ;
225 000056 STATES$
226 000056 TRANS$ <','>
227 ;
228 000056 STATES$
229 000056 TRANS$ $NUMBR,,PVCT ; COUNTER TIMER
230 ;
231 000056 STATES$
232 000056 TRANS$ <','>
233 ;
234 000056 STATES$
235 000056 TRANS$ "OFF",PSZ ; STATE
236 000056 TRANS$ "ON",PSZ
237 ;
238 000056 STATES$ PSZ
239 000056 TRANS$ !END,$EXIT,PVEND
240 000056 TRANS$ <','>
241 ;
242 000056 STATES$
243 000056 TRANS$ $NUMBR,WNDSZ,PVPSZ ; PACKET SIZE
244 000056 TRANS$ $LAMDA
245 ;
246 000056 STATES$ WNDSZ
247 000056 TRANS$ !END,$EXIT,PVEND
248 000056 TRANS$ <','>
249 ;
250 000056 STATES$
251 000056 TRANS$ $NUMBR,OWNER,PVWSZ ; WINDOW SIZE
252 000056 TRANS$ $LAMDA
253 ;
254 000056 STATES$ OWNER
255 000056 TRANS$ !END,$EXIT,PVEND
256 000056 TRANS$ <','>
257 ;
258 000056 STATES$
259 000056 TRANS$ $RAD50,PFLAGS,PVOWN ; PROCESS OWNER
260 000056 TRANS$ $LAMDA

```

CFGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:04 Page 20-1  
PVDLM - CHECK FOR DLM PVC

755 002030 001333  
756 002032  
757

20\$: BNE INVCIR  
RETURN  
.DSABL LSB

```
; BR IF NO
```

★★F

C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C

[illegible]

CEGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:04 Page 21

1000

\*\*FILE\*\*ID\*\*CFGSLT

```

CCCCCCCC  FFFFFFFFFF  GGGGGGGG  SSSSSSSS  LL  TTTTTTTTTT
CCCCCCCC  FFFFFFFFFF  GGGGGGGG  SSSSSSSS  LL  TTTTTTTTTT
CC         FF         GG         SS         LL  TT
CC         FF         GG         SS         LL  TT
CC         FF         GG         SS         LL  TT
CC         FF         GG         SS         LL  TT
CC         FFFFFFFF  GG         SSSSSS  LL  TT
CC         FFFFFFFF  GG         SSSSSS  LL  TT
CC         FF         GG  GGGGGG  SS  LL  TT
CC         FF         GG  GGGGGG  SS  LL  TT
CC         FF         GG  GG      SS  LL  TT
CC         FF         GG  GG      SS  LL  TT
CCCCCCCC  FF         GGGGGG  SSSSSSSS  LLLLLLLLLL  TT
CCCCCCCC  FF         GGGGGG  SSSSSSSS  LLLLLLLLLL  TT

```

```

LL  SSSSSSSS  TTTTTTTTTT
LL  SSSSSSSS  TTTTTTTTTT
LL  SS        TT
LL  SS        TT
LL  SS        TT
LL  SS        TT
LL  SS        TT
LL  SSSSSS    TT
LL  SSSSSS    TT
LL  SS        TT
LL  SS        TT
LL  SS        TT
LL  SS        TT
LL  SS        TT
LL  SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT

```

\*\*FILE\*\*ID\*\*CFGSTA

|          |          |          |          |          |          |    |
|----------|----------|----------|----------|----------|----------|----|
| CCCCCCCC | FFFFFFFF | GGGGGGGG | SSSSSSSS | TTTTTTTT | AAAAAA   |    |
| CCCCCCCC | FFFFFFFF | GGGGGGGG | SSSSSSSS | TTTTTTTT | AAAAAA   |    |
| CC       | FF       | GG       | SS       | TT       | AA       | AA |
| CC       | FF       | GG       | SS       | TT       | AA       | AA |
| CC       | FF       | GG       | SS       | TT       | AA       | AA |
| CC       | FF       | GG       | SS       | TT       | AA       | AA |
| CC       | FFFFFFFF | GG       | SSSSSS   | TT       | AA       | AA |
| CC       | FFFFFFFF | GG       | SSSSSS   | TT       | AA       | AA |
| CC       | FF       | GG       | GGGGGG   | SS       | TTTTTTTT |    |
| CC       | FF       | GG       | GGGGGG   | SS       | TTTTTTTT |    |
| CC       | FF       | GG       | GG       | SS       | TTTTTTTT |    |
| CC       | FF       | GG       | GG       | SS       | TTTTTTTT |    |
| CCCCCCCC | FF       | GGGGGG   | SSSSSSSS | TT       | AA       | AA |
| CCCCCCCC | FF       | GGGGGG   | SSSSSSSS | TT       | AA       | AA |

|            |          |          |
|------------|----------|----------|
| LL         | SSSSSSSS | TTTTTTTT |
| LL         | SSSSSSSS | TTTTTTTT |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SSSSSS   | TT       |
| LL         | SSSSSS   | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LLLLLLLLLL | SSSSSSSS | TT       |
| LLLLLLLLLL | SSSSSSSS | TT       |

| SYMBOL    | VALUE      | REFERENCES                          |
|-----------|------------|-------------------------------------|
| L.UNT     | 000013     | #5-63                               |
| MAXCST    | = 000031   | #6-70                               |
| RTSPC     | = ***** GX | 7-75 10-198                         |
| SCNT      | = ***** GX | 7-76 7-77 7-78 7-79 11-235          |
| SF.ACT    | = 000200   | 10-184 *10-189 10-190 10-205 10-222 |
| SF.ENA    | = 000100   | #5-63                               |
| SF.LPB    | = 000004   | #5-63                               |
| SF.MFL    | = 000040   | #5-63                               |
| SF.PAC    | = 000020   | #5-63                               |
| SF.REA    | = 000010   | #5-63                               |
| SF.SER    | = 000001   | #5-63                               |
| SF.SVC    | = 000002   | #5-63                               |
| SF.UNL    | = 000040   | #5-63                               |
| STADF     | 000000 R   | 8-96                                |
| STAKW     | 000000 RG  | 8-98 #9-114                         |
| STAST     | 000000 RG  | #9-114                              |
| ST.APR    | 000446 R   | #10-219                             |
| ST.CST    | 000356 R   | #10-198                             |
| ST.HTM    | 000500 R   | #11-233                             |
| ST.NUM    | 000270 R   | #10-180                             |
| SYNERR    | = ***** GX | *8-101 *8-104                       |
| \$\$\$BAS | = *****    | 7-75 7-76 7-77 7-78 7-79            |
| S.COST    | 000001     | #5-63 *10-211                       |
| S.FLG     | 000000     | #5-63                               |
| S.LEN     | 000004     | #5-63                               |
| S.NMST    | 000002     | #5-63                               |
| S.OWNR    | 000003     | #5-63                               |
| \$ALPHA   | = 000022   | #9-114                              |
| \$ANY     | = 000020   | #9-114                              |
| \$BLANK   | = 000006   | #9-114                              |
| \$DIGIT   | = 000024   | #9-114                              |
| \$DNUMB   | = 000014   | #9-114                              |
| \$EOS     | = 000012   | #9-114                              |
| \$ERRYA   | 000130 R   | #7-78 11-243                        |
| \$ERRYB   | 000162 R   | #7-79                               |
| \$ERRYP   | = ***** GX | 10-214                              |
| \$ERRYQ   | 000070 R   | 10-227                              |
| \$ERR1A   | 000000 R   | #7-75 10-192                        |
| \$ERR1B   | 000030 R   | #7-76 10-193                        |
| \$ERR1T   | = ***** GX | 8-110                               |
| \$EXIT    | = 000000   | #9-114                              |
| \$FAIL    | = 177777   | #9-114                              |
| \$GPRM    | = *****    | 9-114                               |
| \$LAMDA   | = 000000   | #9-114                              |
| \$NUMBR   | = 000002   | #9-114                              |
| \$QSTA    | 000216 RG  | #8-96                               |
| \$RAD50   | = 000016   | #9-114                              |
| \$RONLY   | = *****    | 9-114 9-114                         |
| \$SLTA    | = ***** GX | 10-200                              |
| \$STRNG   | = 000004   | #9-114                              |
| \$SUBXP   | = 000010   | #9-114                              |



```

304
305
306      ; PACKET SIZE
307
308 000236 005767 000000G  SVPKT: TST .PNUMH      ; LEGAL PACKET SIZE?
309 000242 001004      BNE 101$      ; BR IF NO
310 000244 016767 000000G 000006'  MOV .PNUMB,PkTSVC ; SAVE PACKET SIZE
311 000252      RETURN
312
313      ; ERROR
314
315 000254      101$: MSG$R 3D      ; ILLEGAL PACKET SIZE
316
317      ; WINDOW SIZE
318
319
320 000262 005767 000000G  SVWND: TST .PNUMH      ; LEGAL WINDOW SIZE?
321 000266 001010      BNE 101$      ; BR IF NO
322 000270 022767 000007 000000G  CMP #MAXWND,.PNUMB
323 000276 103404      BLO 101$      ; BR IF ILLEGAL WINDOW SIZE
324 000300 116767 000000G 000022'  MOV .PNUMB,WNDSVC ; SAVE WINDOW SIZE
325 000306      RETURN
326
327      ; ERROR
328
329 000310      101$: MSG$R 3E      ; ILLEGAL WINDOW SIZE
330
331      ; FLAGS WORD
332
333
334 000316 005767 000000G  SVFLAG: TST .PNUMH     ; LEGAL FLAGS WORD?
335 000322 001004      BNE 101$     ; BR IF NO
336 000324 016767 000000G 000010'  MOV .PNUMB,FLGSVC ; SAVE FLAGS WORD
337 000332      RETURN
338
339      ; ERROR
340
341 000334      101$: MSG$R 3F      ; ILLEGAL FLAGS WORD
342
343      ; RECALL TIMER
344
345
346 000342 005767 000000G  SVRCLL: TST .PNUMH     ; LEGAL RECALL TIMER?
347 000346 001007      BNE 101$     ; BR IF NO
348 000350 105767 000001G      TSTB .PNUMB+1
349 000354 001004      BNE 101$     ; BR IF ILLEGAL RECALL TIMER
350 000356 116767 000000G 000023'  MOV .PNUMB,RCLSV ; SAVE RECALL TIMER
351 000364      RETURN
352
353      ; ERROR
354
355 000366      101$: MSG$R 3G      ; ILLEGAL RECALL TIMER
  
```

CFGSVC CREATED BY MACRO ON 29-JUN-85 AT 00:07 PAGE 3 6 14  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                                      |
|---------|------------|---|
| R\$11D  | = *****    | 16-612  |
| SETTLC  | 001202 R   | 13-482 #15-537                                  |
| SF3INC  | = 000002   | 12-372  |
| SF.ACT  | = 000200   | #5-65   |
| SF.ENA  | = 000100   | #5-65   |
| SF.LPB  | = 000004   | #5-65   |
| SF.MFL  | = 000040   | #5-65   |
| SF.PAC  | = 000020   | #5-65   |
| SF.REA  | = 000010   | #5-65   |
| SF.SER  | = 000001   | #5-65   |
| SF.SVC  | = 000002   | #5-65   |
| SF.UNL  | = 000040   | #5-65   |
| SLTA    | 000016 R   | #6-122 *13-426 15-538                           |
| SLTMA   | = ***** GX | 15-537 15-541                                   |
| STASVC  | 000014 R   | #6-121 *10-293 13-427 13-454 15-545             |
| STRNXT  | 001520 R   | 12-389 #17-637                                  |
| SVCDF   | 000000 R   | 8-165   |
| SVCKW   | 000000 RG  | 8-167 #9-185                                    |
| SVCST   | 000000 RG  | #9-185  |
| SVCTL   | 000114 R   | #10-275   |
| SVDIED  | 000476 R   | #12-395   |
| SVDTEE  | 000522 R   | #12-407   |
| SVDTES  | 000440 RG  | #12-382   |
| SVFLAG  | 000316 R   | #11-334   |
| SVNAM   | 000052 R   | #10-261   |
| SVOWNR  | 000550 R   | #13-417   |
| SVPKT   | 000236 R   | #11-308   |
| SVRCLL  | 000342 R   | #11-346   |
| SVRTRY  | 000374 R   | #12-360   |
| SVSTA   | 000164 R   | #10-289   |
| SVWND   | 000262 R   | #11-320   |
| SYNERR  | = ***** GX | *8-173 *10-283 *10-296                          |
| S\$ADSZ | 000007     | *13-460   |
| S\$CTL  | 000002     | *13-453   |
| S\$DTE  | 000020     | 13-463  |
| S\$FLG  | 000010     | *13-456   |
| S\$LEN  | 000032     | 13-436  |
| S\$OWNR | 000030     | *13-468   |
| S\$PKSZ | 000004     | *13-455   |
| S\$RCL  | 000013     | *13-457   |
| S\$STA  | 000003     | *13-454   |
| S\$TMR  | 000016     | *13-459   |
| S\$WND  | 000006     | *13-458   |
| S\$BAS  | = *****    | 7-138 7-139 7-139 7-140 7-140 7-141 7-141 7-142 |
|         |            | 7-142 7-143 7-144 7-144 7-145 7-146 7-146       |
|         |            | 7-147 7-147                                     |
| S.COST  | 000001     | #5-65   |
| S.FLG   | 000000     | #5-65   |
| S.LEN   | 000004     | #5-65   |
| S.NMST  | 000002     | #5-65   |
| S.OWNR  | 000003     | #5-65   |
| T\$FLAG | 000044     | #5-62 *15-572                                   |

```

210      .SBTTL  UNT$DF ACTION ROUTINES
211
212      ;
213      ; UNIT NUMBER (UNT$DF)
214
215      000112  005767  000000G  U.UNIT:  TST      .PNUMH      ; MAKE SURE IT'S A VALID UNIT NUMBER
216      000116  001011      BNE      101$      ;
217      000120  026727  000000G  000060      CMP      .PNUMB,#48.  ; MAX OF 48. ALLOWED
218      000126  103005      BHIS     101$      ;
219      000130  026767  000000G  000000G      CMP      .PNUMB,UCNT  ; IS THIS THE NEXT ASCENDING UNIT NUMBER?
220      000136  001004      BNE      111$      ; IF NE, NO
221      000140      RETURN
222      000142      101$:  MSG$R  1G      ; ILLEGAL UNIT NUMBER
223      000150      111$:  MSG$R  1H      ; UNT$DF OUT OF ORDER
224
225      ;
226      ; CHARACTERISTICS WORD 0 (UNT$DF)
227
228      000156  005767  000000G  U.CH0:  TST      .PNUMH      ; MUST BE SINGLE PRECISION
229      000162  001034      BNE      CHERR      ;
230      000164  016700  000000G      MOV      UCNT,R0      ; GET UNIT COUNT
231      000170  016701  000000G      MOV      $SLTA,R1      ; SLT ADDRESS
232      000174  016702  000000G      MOV      .PNUMB,R2      ; AND CHARACTERISTICS VALUE
233      000200  126100  000013      CMPB     L.UNT(R1),R0  ; IS THIS THE KEY UNIT?
234      000204  001016      BNE      10$      ; IF NE, NO
235      000206  032767  000000G  000000G      BIT      #FL.FDX,$FLAGS  ; FULL-DUPLEX SPECIFIED?
236      000214  001402      BEQ      2$      ; IF EQ, NO
237      000216  042702  000001      BIC      #1,R2      ; BIT 0 OFF=FDX
238      000222  032767  000000G  000000G  2$:  BIT      #FL.HDX,$FLAGS  ; HALF-DUPLEX SPECIFIED?
239      000230  001402      BEQ      5$      ; IF EQ, NO
240      000232  052702  000001      BIS      #1,R2      ; BIT 0 ON=HDX
241      000236  010267  000000G      5$:  MOV      R2,$$DCHA  ; SAVE DLC CHARACTERISTICS
242      000242  006300      10$:  ASL      R0      ; CONVERT TO A DOUBLEWORD INDEX
243      000244  006300      ASL      R0
244      000246  010260  000004G      MOV      R2,$$DCHA+4(R0) ; STORE PARAMETER VALUE
245      000252      RETURN
246      000254      CHERR:  MSG$R  1K      ; ILLEGAL PARAMETER VALUE
247
248      ;
249      ; CHARACTERISTICS WORD 1 (UNT$DF)
250
251      000262  005767  000000G  U.CH1:  TST      .PNUMH      ; MUST BE SINGLE PRECISION
252      000266  001372      BNE      CHERR      ;
253      000270  016700  000000G      MOV      UCNT,R0      ; GET UNIT COUNT
254      000274      CALL      SETTMO      ; SET UP TIMEOUT VALUE
255      000300  016701  000000G      MOV      $SLTA,R1      ; AND SLT ADDRESS
256      000304  126100  000013      CMPB     L.UNT(R1),R0  ; IS THIS THE KEY UNIT?
257      000310  001003      BNE      10$      ; IF NE, NO
258      000312  016767  000000G  000002G      MOV      .PNUMB,$$DCHA+2 ; SAVE DLC CHARACTERISTICS
259      000320  006300      10$:  ASL      R0      ; CONVERT TO A DOUBLEWORD INDEX
260      000322  006300      ASL      R0
261      000324  016760  000000G  000006G      MOV      .PNUMB,$$DCHA+6(R0) ; STORE PARAMETER VALUE
262      000332      RETURN
263
264      ;
265      ; PROTOCOL EMULATOR CHARACTERISTICS WORD
266

```

CFGX2P - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 00:08 <sup>H 16</sup>  
Table of contents

|     |     |                                 |
|-----|-----|---------------------------------|
| 5-  | 55  | MACRO DEFINITIONS               |
| 6-  | 72  | CONSTANT DEFINITIONS            |
| 7-  | 82  | LOCAL DATA                      |
| 8-  | 92  | ERROR MESSAGES                  |
| 9-  | 103 | \$GX2P - LOOK FOR X2P\$DF MACRO |
| 10- | 132 | X2P\$DF STATE TABLE             |
| 11- | 195 | X2P\$DF ACTION ROUTINES         |

|        |        |                  |                  |
|--------|--------|------------------|------------------|
| NNN    | NNN    | TTTTTTTTTTTTTTTT | LLL              |
| NNN    | NNN    | TTTTTTTTTTTTTTTT | LLL              |
| NNN    | NNN    | TTTTTTTTTTTTTTTT | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNNNNN | NNN    | TTT              | LLL              |
| NNNNNN | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNNNNN | TTT              | LLL              |
| NNN    | NNNNNN | TTT              | LLL              |
| NNN    | NNNNNN | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLL              |
| NNN    | NNN    | TTT              | LLLLLLLLLLLLLLLL |
| NNN    | NNN    | TTT              | LLLLLLLLLLLLLLLL |
| NNN    | NNN    | TTT              | LLLLLLLLLLLLLLLL |

ALC  
MAC  
MAC  
NHW  
RET

ALCMEM      CREATED BY    MACRO    ON 29-JUN-85 AT 00:01      PAGE 2      H 2  
MACRO CROSS REFERENCE      CREF    04.00  
MACRO NAME      REFERENCES  
NHWD\$      #5-115      5-117  
RETURN      6-226      7-366

CFG  
MACR

```

53      .SBTTL  MACRO DEFINITIONS
54      :
55      : LIBRARY MACROS
56      :
57      : .MCALL  EMSG$, ISTAT$, STATES$, TRANS$, NTLER$, ASL$
58      :
59      : LOCAL MACROS
60      :
61      :
62      : REJECT TPARS TRANSITION
63      :
64      :
65      : .MACRO  REJ$
66      : ADD    #2,(SP)      : RETURN TO CALLER+2
67      : CLR    SYNERR      : INDICATE NO SYNTAX ERROR
68      : .ENDM  REJ$

```

```
53      .SBTTL  MACRO DEFINITIONS
54
55      : LIBRARY MACROS
56      :
57      .MCALL  EMSG$, ISTAT$, STATE$, TRANS$
58
```



```

LOCAL DATA
98          .SBTTL  LOCAL DATA
99
100         ;
101         ; LOCAL DATA
102         ;
103
104         .PSECT  DATA,D
105
106         .NLIST  BEX
107
108         ;
109         ; LOCAL DATA FOR DTE$DF MACRO
110         ;
111         000017
112         000000
113         000016'
114         000017
115         000006
116         000027
117         000035
118         000036
119
120         000040
121         000042
122         000044
123         000045

DTEADD: .BLKB  DTEALN = 15.      ; MAXIMUM NUMBER OF DIGITS IN DTE ADDRESS
        .BLKB  DTEND = -1      ; TEMPORARY STORAGE FOR UNPACKED DTE ADDRESS
DTEPCK: .BLKB  8.              ; PACKED DTE ADDRESS
        .BLKB  DTNMMX = 6.     ; MAX CHARACTERS IN NETWORK NAME
DTNTNM: .BLKB  DTNMMX          ; NETWORK NAME
COUNT: .BLKB  1              ; COUNT OF NUMBER OF DIGITS IN DTE ADDRESS
CVTBUF: .BLKB  1              ; CONVERSION BUFFER FOR ASCII TO DECIMAL DIGIT
        .EVEN
NEXT:   .BLKW  1              ; ADDRESS OF NEXT AVAILABLE BYTE
LINNAM: .BLKW  1              ; DEVICE NAME FROM LINE-ID (RAD50)
LINCTL: .BLKB  1              ; CONTROLLER NUMBER
LINUNT: .BLKB  1              ; UNIT NUMBER

```

```

473 .SBTTL PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
474 +
475 PCKBCD - PACK A STRING OF DIGITS IN BCD FORMAT
476
477 INPUTS:
478 R0 - NUMBER OF BYTES IN PACKED ADDRESS
479 R4 - ADDRESS OF BUFFER CONTAINING ASCII DIGITS TO PACK (0=END OF BUFFER)
480 R5 - ADDRESS OF BUFFER TO STORE PACKED DATA
481 COUNT - NUMBER OF DIGITS SPECIFIED
482
483 OUTPUTS:
484 BUFFER SPECIFIED IN R5 CONTAINS DATA IN BCD FORMAT
485
486
487 -
488
489 000660 010501 PCKBCD: MOV R5,R1 ; GET ADDRESS OF PACKED DATA
490 000662 116702 MOV COUNT,R2 ; GET NUMBER OF DIGITS IN STRING
491 000666 001423 BEQ 40$ ; BR IF NOTHING TO DO
492 000670 105021 5$: CLRB (R1)+ ; INITIALIZE PACKED ADDRESS
493 000672 005300 DEC R0 ; MORE TO INITIALIZE?
494 000674 003375 BGT 5$ ; BR IF YES
495 000676 005001 CLR R1 ; POSITION INIDCATOR (0=HIGH FOUR BITS)
496 000700 005305 DEC R5 ; UPDATE ADDRESS FOR FIRST TIME
497 000702 112400 10$: MOVB (R4)+,R0 ; GET DIGIT TO PACK
498 000704 005701 TST R1 ; STORE IN HIGH FOUR BITS?
499 000706 001007 BNE 20$ ; BR IF NO
500 000710 ASL$ 4,R0 ; MOVE TO HIGH FOUR BITS
501 000720 005201 INC R1 ; INDICATE STORE IN LOW FOUR BITS NEXT
502 000722 005205 INC R5 ; POINT TO NEXT BYTE
503 000724 000401 BR 30$ ; BR TO STORE DECIMAL
504 000726 005301 20$: DEC R1 ; INDICATE STORE IN HIGH FOUR BITS NEXT
505 000730 150015 30$: BISB R0,(R5) ; PACK THE NUMBER INTO THE BUFFER
506 000732 005302 DEC R2 ; MORE TO CONVERT?
507 000734 003362 BGT 10$ ; BR IF YES
508 000736 000001 40$: RETURN
509 000001 .END

```

```

140      .SBTTL  FEA$DF ACTION ROUTINES
141
142      ; FIRST CHARACTER OF DEVICE NAME (FEA$DF)
143
144 000052 126767 000000G 000000G F.CH1: CMPB  .PCHAR,$$DEV ; IS THIS THE RIGHT FEATURES DEFINITION
145 000060 001404          BEQ    10$ ; IF EQ, YES
146 000062          REJ$ ; ELSE, REJECT TRANSITION
147 000072          10$: RETURN
148
149      ; SECOND CHARACTER OF DEVICE NAME (FEA$DF)
150
151 000074 126767 000000G 000001G F.CH2: CMPB  .PCHAR,$$DEV+1 ; ARE WE SURE THIS IS THE RIGHT ONE ?
152 000102 001404          BEQ    10$ ; IF EQ, YUP
153 000104          REJ$ ; ELSE, REJECT
154 000114          10$: RETURN
155
156      ; INITIAL FEATURES WORD (FEA$DF)
157
158 000116 016767 000000G 000000G F.FEA: MOV  .PNUMB,$FEATR ; SETUP THE FEATURES WORD
159 000124          RETURN
160
161      ; MODIFIABLE BITS IN THE FEATURES WORD (FEA$DF)
162
163 000126 046767 000000G 000000G F.MOD: BIC  .PNUMB,$FEATR ; CLEAR THE MODIFIABLE BITS
164 000134          RETURN
165
166      ; BITS CURRENTLY SET IN THE FEATURES WORD (FEA$DF)
167
168 000136 056767 000000G 000000G F.SET: BIS  .PNUMB,$FEATR ; SET THE CURRENT STATES
169 000144          RETURN
170
171
172      .END

```

```

261
262 000056          STATES$ PFLAGS
263 000056          TRANS$ !END,$EXIT,PVEND
264 000056          TRANS$ <','>
265
266 000056          STATES$
267 000056          TRANS$ $NUMBR,END,PVFG          ; FLAGS BYTE
268 000056          TRANS$ $LAMDA
269
270 000056          STATES$ END
271 000056          TRANS$ $EOS,$EXIT
272 000056          TRANS$ <','>,$EXIT
273
274          ; CHECK FOR DLM CIRCUIT NAME - SUBEXPRESSION
275          ;
276 000056          STATES$ CHKDLM
277 000056          TRANS$ '-,,PVDLM
278
279 000056          STATES$
280 000056          TRANS$ $DNUMB,,PVCTL
281
282 000056          STATES$
283 000056          TRANS$ '
284
285 000056          STATES$
286 000056          TRANS$ $DNUMB,,PVSTA,PF.DLM,FLAG
287
288 000056          STATES$
289 000056          TRANS$ <','>,$EXIT
290
291 000056          STATES$

```

```

759 .SBTTL DLMTLC - SET UP TRANSPORT LINE COUNTERS
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775 002034 017703 000000G DLMTLC: MOV @SLTMA,R3 :: GET ADDRESS OF VECTOR TABLE
776 002040 022367 000000G 5$: CMP (R3)+,$SLTA :: IS THIS THE CORRECT ENTRY?
777 002044 001375 BNE 5$ :: BR IF NO
778 002046 005743 TST -(R3) :: BACK UP TO CORRECT ENTRY
779 002050 167703 000000G SUB @SLTMA,R3 :: GET SYSTEM LINE NUMBER*2
780 002054 010367 000030' MOV R3,DLMSLN :: SAVE SYSTEM LINE NUMBER FOR LATER
781 002060 006267 000030' ASR DLMSLN
782 002064 067703 000000G ADD @LLCTA,R3 :: INDEX INTO REVERSE MAPPING TABLE
783 002070 011303 MOV (R3),R3 :: GET CHANNEL NUMBER AND PDV INDEX
784 002072 100004 BPL 10$ :: BR IF NOT MULTIPOINT
785 002074 066703 000026' ADD DLMSTA,R3 :: POINT TO CORRECT STATION
786 002100 006303 ASL R3 :: GET ADDRESS OF STATION TABLE
787 002102 011303 MOV (R3),R3 :: GET CHANNEL NUMBER AND PDV INDEX
788 002104 005046 10$: CLR -(SP) :: GET INDEX INTO FORWARD MAPPING TABLE
789 002106 110316 MOVB R3,(SP) :: ...
790 002110 006316 ASL (SP) :: ...
791 002112 012603 MOV (SP)+,R3 :: RETRIEVE FORWARD MAPPING TABLE INDEX
792 002114 017704 000000G MOV @PDVTA,R4 :: GET ADDRESS OF PDV VECTOR TABLE
793 002120 017700 000000G MOV @PDVNM,R0 :: GET NUMBER OF PDV'S IN SYSTEM
794 002124 012402 15$: MOV (R4)+,R2 :: GET NEXT PDV IN LIST
795 002126 022762 114224 000000G CMP #*RXP1,Z.NAM(R2) :: IS THIS XPT?
796 002134 001403 BEQ 17$ :: BR IF YES
797 002136 SOB R0,15$ :: CONTINUE UNTIL ALL CHECKED
798 002142 000452 BR 60$ :: XPT NOT IN SYSTEM?
799 002144 060203 17$: ADD R2,R3 :: POINT TO FORWARD MAPPING TABLE ENTRY
800 002146 062703 000000G ADD #Z.MAP,R3
801 002152 011304 MOV (R3),R4
802 002154 SAVMAP
803 002160 016202 000000G MOV Z.DAT(R2),R2
804 002164 MAP N$TLC+2(R2)
805 002172 016203 000100 MOV N$TLC(R2),R3
806 002176 016202 000104 MOV N$TLC+4(R2),R2
807 002202 BIAS R2
808 002212 005712 20$: TST (R2)
809 002214 100407 BMI 30$
810 002216 021204 CMP (R2),R4
811 002220 001406 BEQ 40$
812 002222 062702 000046 ADD #ISLLEN,R2
813 002226 SOB R3,20$
814 002232 000414 BR 50$
815 002234 010412 30$: MOV R4,(R2)

```

CFGSLT - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 <sup>H 10</sup> 00:05  
Table of contents

|    |     |                                 |
|----|-----|---------------------------------|
| 5- | 52  | MACRO DEFINITIONS               |
| 6- | 71  | \$QSLT - LOOK FOR SLT\$DF MACRO |
| 7- | 103 | TPARS TABLES                    |
| 8- | 164 | SUBEXPRESSIONS                  |
| 9- | 188 | ACTION ROUTINES                 |

CFGSTA - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 00:06 <sup>H 11</sup>  
Table of contents

|     |     |                           |
|-----|-----|---------------------------|
| 5-  | 57  | MACRO DEFINITIONS         |
| 6-  | 65  | CONSTANT DEFINITIONS      |
| 7-  | 72  | ERROR MESSAGES            |
| 8-  | 82  | \$QSTA - LOOK FOR STA\$DF |
| 9-  | 112 | STA\$DF STATE TABLE       |
| 10- | 175 | STA\$DF ACTION ROUTINES   |

CFGSTA  
SYME  
SYME  
\$\$AF  
\$\$H  
\$\$L  
\$\$M  
\$\$N  
\$\$S  
\$\$S  
\$.PNU  
\$.PNU  
\$.TP

CFGSTA      CREATED BY    MACRO    ON 29-JUN-85 AT 00:06      PAGE 3      H 12

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL    | VALUE      | REFERENCES  |
|-----------|------------|---|
| \$\$APR   | = ***** GX | *10-224   |
| \$\$HTIM  | = ***** GX | *11-238   |
| \$\$LTIM  | = ***** GX | *11-240   |
| \$\$MTP   | = ***** GX | *10-190   |
| \$\$SNUM  | = ***** GX | *10-187   |
| \$\$\$FLG | = 177777   | #9-114  |
| \$\$\$KEY | = 177777   | #9-114  |
| .PNUMB    | = ***** GX | 10-182      10-187      10-198      10-211      10-220      10-224      11-238      *11-239      11-240 |
| .PNUMH    | = ***** GX | 10-180      11-233  |
| .TPARS    | = ***** GX | 8-102   |

CFG  
SVC



```

357      ; RETRY TIMER
358
359      ;
360 000374 005767 000000G SVRTY: TST      ,PNUMH      ; LEGAL RETRY TIMER?
361 000400 001004      BNE      101$      ; BR IF NO
362 000402 016767 000000G 000012'      MOV      ,PNUMB,RTYSVC ; SAVE RETRY TIMER
363 000410      RETURN
364
365      ; ERROR
366
367 000412      101$: MSG$R 3H      ; ILLEGAL RETRY TIMER
368
369      ;
370      ; NO DTE ADDRESS SPECIFIED
371
372 000420 032767 000002 000010' NODTE: BIT      ,SF$INC,FLGSVC ; IS THIS AN INCOMING SVC?
373 000426 001401      BEQ      101$      ; BR IF NO
374 000430      RETURN
375
376      ; ERROR
377
378 000432      101$: MSG$R 3M      ; DTE ADDRESS MISSING
379
380      ; START OF DTE ADDRESS
381
382 000440 012700 000017' SVDTES: MOV      ,DTELEN,R0      ; GET LENGTH OF REMOTE DTE ADDRESS
383 000444 012701 000025'      MOV      ,DTEVC,R1      ; GET ADDRESS OF START OF BUFFER
384 000450 105021      10$: CLRB      (R1)+      ; INITIALIZE BUFFER
385 000454 005300      DEC      R0      ; MORE TO INITIALIZE?
386 000458 003375      BGT      10$      ; BR IF YES
387 000462 105067 000024'      CLRB      COUNT      ; INITIALIZE DIGIT COUNT
388 000466 012767 000025' 000000'      MOV      ,DTEVC,NEXT ; START AT BEGINNING OF BUFFER
389 000470      CALL      STRNXT      ; STORE CHARACTER
390 000474      RETURN
391
392      ;
393      ; PROCESS DTE DIGIT
394
395 000476 026727 000000' 000044' SVDTED: CMP      NEXT,#DTEEND ; IS ADDRESS TOO LONG?
396 000504 101033      BHI      101$      ; BR IF YES
397 000506      CALL      STRNXT      ; STORE CHARACTER
398 000512      RETURN
399
400      ; ERRORS
401
402 000514      101$: MSG$R 3I      ; ILLEGAL DTE ADDRESS
403
404      ;
405      ; END OF DTE ADDRESS
406
407 000522 004567 000000G SVDTEE: JSR      R5,$$AVRG      ; SAVE R3-R5
408 000526 012700 000010'      MOV      ,#<DTELEN+1>/2,R0 ; GET LENGTH OF PACKED ADDRESS
409 000532 012704 000025'      MOV      ,DTEVC,R4      ; POINT TO NUMBER TO CONVERT
410 000536 012705 000044'      MOV      ,DTEPCK,R5 ; POINT TO STORAGE FOR PACKED NUMBER
411 000542      CALL      PCKBCD      ; PACK REMOTE DTE ADDRESS IN BCD
412 000546      RETURN

```

CFGSVC CREATED BY MACRO ON 29-JUN-85 AT 00:07 PAGE 4 H 14  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL   | VALUE      | REFERENCES                    |
|----------|------------|-------------------------------|
| T\$LIF   | 000013     | #5-62                         |
| T\$LIFL  | 000013     | #5-62                         |
| T\$LIFO  | 000013     | #5-62                         |
| T\$LIFS  | 000013     | #5-62                         |
| T\$LIN   | 000000     | #5-62                         |
| T\$LIPS  | 000006     | #5-62                         |
| T\$LLD   | 000012     | #5-62                         |
| T\$LLDC  | 000045     | #5-62                         |
| T\$LLDL  | 000012     | #5-62                         |
| T\$LLDO  | 000012     | #5-62                         |
| T\$LLDS  | 000012     | #5-62                         |
| T\$LLEN  | 000046     | #5-62 15-569                  |
| T\$LOPR  | 000002     | #5-62                         |
| T\$LTCL  | 000024     | #5-62                         |
| T\$LTIM  | 000026     | #5-62                         |
| T\$LTIPR | 000014     | #5-62                         |
| T\$LTIPS | 000020     | #5-62                         |
| T\$NAPL  | 000004     | #5-62                         |
| T\$NFE   | 000000     | #5-62                         |
| T\$NLEN  | 000010     | #5-62                         |
| T\$NNUL  | 000002     | #5-62                         |
| T\$NOPL  | 000006     | #5-62                         |
| T\$NRNI  | 000042     | #5-62                         |
| T\$NRPL  | 000005     | #5-62                         |
| T\$NRUL  | 000007     | #5-62                         |
| T\$NVR   | 000001     | #5-62                         |
| T\$RPRI  | 000040     | #5-62                         |
| T\$SVC   | 000034     | #5-62 *15-573 *15-578 *15-579 |
| T\$T5    | 000030     | #5-62 *15-580                 |
| T\$T6    | 000032     | #5-62 *15-581                 |
| WINDSVC  | 000022 R   | #6-125 *10-269 *11-324 13-458 |
| Z.DAT    | = ***** GX | 15-560                        |
| Z.MAP    | = ***** GX | 15-557                        |
| \$ALPHA  | = 000022   | #9-185                        |
| \$ANY    | = 000020   | #9-185                        |
| \$BLANK  | = 000006   | #9-185                        |
| \$CAT5   | = ***** GX | 10-265 13-421                 |
| \$CDTB   | = ***** GX | 17-639                        |
| \$CEACX  | = ***** GX | 13-445 13-474                 |
| \$DIGIT  | = 000024   | #9-185                        |
| \$DNUMB  | = 000014   | #9-185                        |
| \$EOS    | = 000012   | #9-185                        |
| \$ERR1T  | = ***** GX | 8-179                         |
| \$ERR2I  | = ***** GX | 10-301                        |
| \$ERR3D  | 000056 R   | #7-138 11-315                 |
| \$ERR3E  | 000102 R   | #7-139 11-329                 |
| \$ERR3F  | 000126 R   | #7-140 11-341                 |
| \$ERR3G  | 000152 R   | #7-141 11-355                 |
| \$ERR3H  | 000200 R   | #7-142 12-367                 |
| \$ERR3I  | 000224 R   | #7-143 12-402                 |
| \$ERR3J  | 000250 R   | #7-144 13-491                 |
| \$ERR3K  | 000266 R   | #7-145 13-493                 |

|     |        |        |                 |             |                       |   |                          |
|-----|--------|--------|-----------------|-------------|-----------------------|---|--------------------------|
| 267 | 000334 | 005767 | 000000G         | U.PECH: TST | .PNUMH                | : | MUST BE SINGLE PRECISION |
| 268 | 000340 | 001345 |                 | BNE         | CHERR                 | : |                          |
| 269 | 000342 | 016700 | 000000G         | MOV         | UCNT,R0               | : | GET UNIT COUNT           |
| 270 | 000346 | 016701 | 000000G         | MOV         | \$SLT,R1              | : | AND SLT ADDRESS          |
| 271 | 000352 | 126100 | 000013          | CMPE        | L.UNIT(R1),R0         | : | IS THIS THE KEY UNIT?    |
| 272 | 000356 | 001003 |                 | BNE         | 10\$                  | : | IF NE, NO                |
| 273 | 000360 | 016767 | 000000G 000000G | MOV         | .PNUMB,\$\$PCHA       | : | SAVE DLC CHARACTERISTICS |
| 274 | 000366 | 006300 | 10\$:           | ASL         | R0                    | : | CONVERT TO A WORD INDEX  |
| 275 | 000370 | 016760 | 000000G 000002G | MOV         | .PNUMB,\$\$PCHA+2(R0) | : | STORE PARAMETER VALUE    |
| 276 | 000376 |        |                 | RETURN      |                       |   |                          |

.TITLE CFGX2P - CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

#### MODULE DESCRIPTION:

NLT - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

#### IDENT HISTORY:

1.00 02-JUN-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1  
  
3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1  
  
4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0  
  
5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

\*\*FILE\*\*ID\*\*ALCMEM

I 1

```
AAAAAA LL CCCCCCCC MM MM EEEEEEEEE MM MM
AAAAAA LL CCCCCCCC MM MM EEEEEEEEE MM MM
AA AA LL CC MMMM MMMM EE MMMM MMMM
AA AA LL CC MMMM MMMM EE MMMM MMMM
AA AA LL CC MM MM EE MM MM
AA AA LL CC MM MM EE MM MM
AA AA LL CC MM MM EEEEEEE MM MM
AAAAAAA LL CC MM MM EEEEEEE MM MM
AAAAAAA LL CC MM MM EEEEEEE MM MM
AA AA LL CC MM MM EE MM MM
AA AA LL CC MM MM EE MM MM
AA AA LL LLLLLLLLL CCCCCCCC MM MM EEEEEEEEE MM MM
AA AA LL LLLLLLLLL CCCCCCCC MM MM EEEEEEEEE MM MM
```

```
LL SSSSSSSS TTTTTTTTT
LL SSSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLL SSSSSSSS TT
LLLLLLLLL SSSSSSSS TT
```

\*\*FILE\*\*ID\*\*CEALL

```

CCCCCCCC  EEEEEEEEE  AAAAAA  LL      LL
CCCCCCCC  EEEEEEEEE  AAAAAA  LL      LL
CC         EE         AA      AA  LL      LL
CC         EE         AA      AA  LL      LL
CC         EE         AA      AA  LL      LL
CC         EE         AA      AA  LL      LL
CC         EEEEEEEE  AA      AA  LL      LL
CC         EEEEEEEE  AA      AA  LL      LL
CC         EE         AAAAAAAAAA LL      LL
CC         EE         AAAAAAAAAA LL      LL
CC         EE         AA      AA  LL      LL
CC         EE         AA      AA  LL      LL
CC         EEEEEEEE  AA      AA  LLLLLLLLLL LLLLLLLLLL
CCCCCCCC  EEEEEEEEE  AA      AA  LLLLLLLLLL LLLLLLLLLL
CCCCCCCC  EEEEEEEEE  AA      AA  LLLLLLLLLL LLLLLLLLLL

```

```

....
....
....
....

```

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LL          SSSSSS    TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT

```

```

70                                     .SBTTL  ERROR MESSAGES
71                                     :
72                                     : ERROR MESSAGES
73                                     :
74 000000                             .PSECT  DATA,D
75                                     .ENABL  LC
76
77 000000                             NTLERS$ ,1C,8B,CERR,RTSPC,CFLIN,<Interrupt Vector>
78 000032                             NTLERS$ ,1D,8,CERR,RTSPC,CFLIN,<Vector Not In System>
79 000070                             NTLERS$ ,1E,8B,CERR,RTSPC,CFLIN,<CSR Address>
80 000114                             NTLERS$ ,1F,8B,CERR,RTSPC,CFLIN,<Interrupt Priority>
81 000150                             NTLERS$ ,72,8A,CERR,REP8A,CFLIN,<Offline>
82                                     .IF DF R$MPL
83                                     NTLERS$ ,7K,8A,CERR,REP8A,CFLIN,<Unibus Run Mask Missing>
84                                     .ENDC
85
86                                     .DSABL  LC
87                                     .EVEN
88                                     .DSABL  LC
89 000000                             .PSECT

```

```

60                                     .SBTTL $QDDM - LOOK FOR DDM$DF MACRO
61
62                                     ;+
63                                     ; $QDDM - LOOK FOR DDM$DF MACRO
64                                     ;
65                                     ; INPUTS:
66                                     ;     NONE
67                                     ;
68                                     ; OUTPUTS:
69                                     ;     C-BIT=SUCCESS/FAILURE
70                                     ;     R3,R4,R5=DESTROYED
71                                     ; -
72
73 000000 012705 000000' $QDDM:: MOV #DD MDF,R5 ; STATE TABLE ADDRESS
74 000004 005001          CLR R1 ; FULL KEYWORD MATCH LENGTH
75 000006 012702 000000' MOV #DD MKW,R2 ; KEYWORD TABLE ADDRESS
76 000012 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
77 000016 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
78 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
79 000026          CALL .TPARS ; GO DO THE PARSE
80 000032 103003          BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
81 000034 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
82 000040 001401          BEQ 101$ ; IF EQ, YES
83 000042          20$: RETURN
84
85                                     ;
86                                     ; ERROR CONDITION
87
88 000044          101$: MSG$R 1T ; SYNTAX ERROR
  
```



ERROR MESSAGES

```
125          .SBTTL  ERROR MESSAGES
126
127          .ENABL  LC
128
129          NTLER$ ,00,8B,CERR,RTSPC,CFLIN,<network name parameter>
130          NTLER$ ,03,8B,CERR,RTSPC,CFLIN,<DTE address value>
131          NTLER$ ,04,8B,CERR,RTSPC,CFLIN,<hash table size>
132          NTLER$ ,05,8B,CERR,RTSPC,CFLIN,<counter timer value>
133          NTLER$ ,06,8B,CERR,RTSPC,CFLIN,<line-id>
134
135          .DSABL  LC
136          .LIST   BEX
137          .EVEN
138          .PSECT
```

|                  |                       |                    |                    |                   |
|------------------|-----------------------|--------------------|--------------------|-------------------|
| ASSCHK= 000000   | HSHMN = 000040        | LINID1 000146R     | 003 L\$NRFS 000100 | P\$P45= 000000    |
| ASSCPS= 000000   | HSHMX = 001000        | LINNAM 000042R     | 002 L\$NRPK 000056 | P\$WRD= 000000    |
| ASSPRI= 000000   | HSHSZ = ***** GX      | LINUNT 000045R     | 002 L\$NRRE 000105 | Q\$DPT= 000010    |
| ASSTRP= 000000   | HSHSTBL 000214R       | LNSOFF= 000001     | L\$NRST 000107     | RTSPC = ***** GX  |
| BIASX 000500R    | HSCUG 000010          | LNSON = 000000     | L\$NTBY 000052     | RSR0 = 000002     |
| CERR = ***** GX  | HSDST 000012          | LNSSHU= 000002     | L\$NTCA 000070     | RSR1 = 000004     |
| CFGBF = ***** GX | HSD29 000014          | LN.CLO= 000000     | L\$NTFS 000102     | RSR2 = 000006     |
| CFGSZ = ***** GX | H\$FLG 000000         | LN.DUM= 000005     | L\$NTPK 000062     | RSR3 = 000010     |
| CFLIN = ***** GX | H\$GLEN 000104        | LN.LOA= 000004     | L\$DMST 000002     | RSR4 = 000012     |
| CHKDTE 000360R   | H\$GLT 000044         | LN.LDD= 000003     | L\$PCS 000112      | RSR5 = 000014     |
| CNTTIM 000264R   | H\$GNAM 000050        | LN.DAU= 000003     | L\$PLS 000110      | R\$SDER= 000000   |
| COUNT 000035R    | 002 H\$GNML= 000020   | LN.OFF= 000001     | L\$QCNH 000122     | R\$SK11= 000001   |
| CVTBUF 000036R   | 002 H\$GPT 000046     | LN.ON = 000000     | L\$QUEH 000124     | R\$SSND= 000000   |
| C\$CKP= 000000   | H\$HITS 000034        | LN.OOP= 000004     | L\$QUET 000130     | R\$S11M= 000000   |
| C\$DRE= 000400   | H\$HLEN 000044        | LN.DPE= 000001     | L\$RTRY 000007     | SF.ACT= 000200    |
| C\$RSH= 177564   | H\$LBDA 000070        | LN.REF= 000002     | L\$SLN 000004      | SF.ENA= 000100    |
| DEVCTL 000546R   | H\$LBDM 000072        | LN.SER= 000002     | L\$ST 000005       | SF.LPB= 000004    |
| DEVNAM 000520R   | H\$LDTE 000002        | LN.STA= 000017     | L\$TCLZ 000044     | SF.MFL= 000040    |
| DEVUNT 000564R   | H\$LEN 000042         | LN.SUB= 000360     | L\$TIM 000006      | SF.PAC= 000020    |
| DE.DFF= ***** GX | H\$LDTS 000032        | LN.TRI= 000006     | L\$SASG= 000000    | SF.REA= 000010    |
| DE.DN = ***** GX | H\$NETW 000024        | LP\$EIP= 002000    | L\$SDRV= 000000    | SF.SER= 000001    |
| DTEAD 000012R    | 003 H\$NML = 000006   | LP\$ENB= 004000    | L\$SP11= 000001    | SF.SVC= 000002    |
| DTEADD 000000R   | 002 H\$NPT 000022     | LP\$PCT= 001400    | L\$S11R= 000000    | SF.UNL= 000040    |
| DTEAEN 000140R   | H\$PTB 000020         | LP\$PMR= 100000    | L.COST 000015      | SPACE = 000040    |
| DTEALN= 000017   | H\$PVC 000006         | LP\$UP = 010000    | L.CTL 000012       | STRNXT 000610R    |
| DTEAST 000052R   | H\$RDTE 000004        | LP\$WTD= 020000    | L.CVA 177776       | SYNERR= ***** GX  |
| DTEDES= ***** GX | H\$RNW 000042         | LP\$WTS= 040000    | L.DDM 000002       | S\$WGRG= 000000   |
| DTEDF 000000R    | 003 H\$SVC 000036     | L\$SGDN= 000004    | L.DDS 000004       | S\$YSZ= 007600    |
| DTEDIG 000114R   | H\$TRB 000016         | L\$SOFF= 000000    | L.DLC 000003       | S.COST 000001     |
| DTEEND 000342R   | H\$XAVL 000100        | L\$SSTR= 000002    | L.DLM 000006       | S.FLG 000000      |
| DTEFLG= ***** GX | H\$XBIA 000074        | L\$SUP = 000003    | L.DLS 000010       | S.LEN 000004      |
| DTEKW 000000RG   | 004 H\$X29C 000040    | L\$SWT = 000001    | L.FLG 000000       | S.NMST 000002     |
| DTEND = 000016R  | 002 I\$SRAR= 000000   | L\$ACHN 000012     | L.KRBA 000016      | S.OWNR 000003     |
| DTENET 000302R   | I\$SRDN= 000000       | L\$APVC 000014     | L.LEN = 000022     | T\$KMG= 000000    |
| DTEW 000076R     | 003 K\$AR5 = ***** GX | L\$ASVC 000010     | L.MPF 000022       | T\$MIN= 000000    |
| DTEPCK 000017R   | 002 K\$CNT= 177546    | L\$AUC 000042      | L.NMST 000020      | V\$CTR= 001000    |
| DTEST 000000RG   | 003 K\$CSR= 177546    | L\$CHLS 000034     | L.NSTA 000014      | X\$DBT= 000000    |
| DINMMX= 000006   | K\$SLDC= 000000       | L\$CHTB 000030     | L.OWNR 000021      | \$ALPHA= 000022   |
| DINTNM 000027R   | 002 K\$STPS= 000074   | L\$CTEN= 000044    | L.UNT 000013       | \$ANY = 000020    |
| D\$BUG= 177514   | LD\$LP = 000000       | L\$CTEN 000032     | M\$SCRB= 000124    | \$BLANK= 000006   |
| D\$ISK= 000000   | LF.ACT= 100000        | L\$CTIM 000040     | M\$SCRX= 000000    | \$CAT5 = ***** GX |
| D\$LL1= 000001   | LF.BRD= 000400        | L\$DTEA 000020     | M\$SFCS= 000000    | \$CDBT = ***** GX |
| D\$SYNC= 000000  | LF.BWT= 000007        | L\$DTEL 000017     | M\$SMGE= 000000    | \$CEACX= ***** GX |
| D\$SYNM= 000000  | LF.ENA= 002000        | L\$GLEN 000134     | M\$SNET= 000000    | \$DIGIT= 000024   |
| END 000114R      | 003 LF.LPB= 001000    | L\$LEN 000122      | M\$SQVR= 000000    | \$DNUMB= 000014   |
| E\$EXPR= 000000  | LF.MDC= 000100        | L\$LLCH 000016     | NEXT 000040R       | \$EOS = 000012    |
| FM78B = ***** GX | LF.MFL= 004000        | L\$LNK 000000      | NUMDTE= 000020     | \$ERRDD 000046R   |
| FM.8B = 000000   | LF.MTP= 000020        | L\$MCHN 000036     | N\$SACC= 000001    | \$ERRD3 000106R   |
| F\$LLVL= 000001  | LF.PAC= 000200        | L\$NETW 000114     | N\$SBUF= 000001    | \$ERRD4 000140R   |
| G\$STPP= 000000  | LF.RDY= 040000        | L\$NIRE 000072     | N\$SLDV= 000001    | \$ERRD5 000170R   |
| G\$STSS= 000000  | LF.REA= 010000        | L\$NLRE 000104     | N\$SMCP= 000001    | \$ERRD6 000224R   |
| G\$STTK= 000000  | LF.SER= 000040        | L\$NMAC 000076     | N\$SMLL= 000001    | \$ERRIT= ***** GX |
| G\$SWRD= 000000  | LF.TIM= 000010        | L\$NMAS 000074     | N\$SMDV= 000010    | \$EXIT = 000000   |
| H\$DLM= 000002   | LF.UNL= 020000        | L\$NMST 000003     | N\$SNCT= 000001    | \$FAIL = 177777   |
| H\$GWY= 000010   | LF.X2P= 000000        | L\$NNRE 000106     | N\$SPEM= 000001    | \$HEADR= ***** GX |
| H\$HOS= 000004   | LINCTL 000044R        | 002 L\$NRBY 000046 | PCKBCD 000660R     | \$LAMD= 000000    |
| H\$XDF= 000020   | LIND 000124R          | 003 L\$NRCA 000066 | PSIPT = ***** GX   | \$NUMBER= 000002  |

|                    |                  |                  |                  |                   |
|--------------------|------------------|------------------|------------------|-------------------|
| A\$\$CHK= 000000   | F.CH1 000052R    | L\$\$P11= 000001 | Q\$\$OPT= 000010 | \$ERR1T= ***** GX |
| A\$\$CPS= 000000   | F.CH2 000074R    | L\$\$1TR= 000000 | R\$\$DER= 000000 | \$EXIT = 000000   |
| A\$\$PRI= 000000   | F.FEA 000116R    | M\$\$CRB= 000124 | R\$\$K11= 000001 | \$FAIL = 177777   |
| \$\$TRP= 000000    | F.MOD 000126R    | M\$\$CRX= 000000 | R\$\$SND= 000000 | \$FEATR= ***** GX |
| CFGBF = ***** GX   | F.SET 000136R    | M\$\$FCS= 000000 | R\$\$11M= 000000 | \$LAMDA= 000000   |
| CFGSZ = ***** GX   | G\$\$TTP= 000000 | M\$\$MGE= 000000 | SYNERR= ***** GX | \$NUMBR= 000002   |
| C\$\$CKP= 000000   | G\$\$TSS= 000000 | M\$\$NET= 000000 | S\$\$WRG= 000000 | \$QFEA 000000RG   |
| C\$\$ORE= 000400   | G\$\$TTK= 000000 | M\$\$OVR= 000000 | S\$\$YSZ= 007600 | \$RAD50= 000016   |
| C\$\$RSH= 177564   | G\$\$WRD= 000000 | N\$\$ACC= 000001 | T\$\$KMG= 000000 | \$STRNG= 000004   |
| D\$\$BUG= 177514   | I\$\$RAR= 000000 | N\$\$BUF= 000001 | T\$\$MIN= 000000 | \$SUBXP= 000010   |
| D\$\$ISK= 000000   | I\$\$RDN= 000000 | N\$\$LDV= 000001 | V\$\$CTR= 001000 | \$SDEV = ***** GX |
| D\$\$L11= 000001   | K\$\$CNT= 177546 | N\$\$MCP= 000001 | X\$\$DBT= 000000 | \$\$FLG= 177777   |
| D\$\$YNC= 000000   | K\$\$CSR= 177546 | N\$\$MML= 000001 | \$ALPHA= 000022  | \$\$KEY= 000000   |
| D\$\$YNM= 000000   | K\$\$LDC= 000000 | N\$\$MOV= 000010 | \$ANY = 000020   | \$\$STA= 000000   |
| E\$\$XPR= 000000   | K\$\$TPS= 000074 | N\$\$NCT= 000001 | \$BLANK= 000006  | \$\$TMP= 000000R  |
| FEADF 000000R 002  | LD\$LP = 000000  | N\$\$PEM= 000001 | \$DIGIT= 000024  | .PCHAR= ***** GX  |
| FEAKW 000000RG 003 | L\$\$ASG= 000000 | P\$\$P45= 000000 | \$DNUMB= 000014  | .PNUMB= ***** GX  |
| FEAST 000000RG 002 | L\$\$DRV= 000000 | P\$\$WRD= 000000 | \$EOS = 000012   | .TPARS= ***** GX  |
| F\$\$LVL= 000001   |                  |                  |                  |                   |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
000146 001 (RW,I,LCL,REL,CON)  
\$STATE 000042 002 (RW,D,LCL,REL,CON)  
\$KTAB 000002 003 (RW,D,LCL,REL,CON)  
\$KSTR 000007 004 (RW,D,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 11377 Words ( 45 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:18.13  
SY:CFGFEA.V2,[132,134]CFGFEA/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFGFEA

```

293 .SBTTL PVC$DF ACTION ROUTINES
294
295 ; PVC CIRCUIT ID
296
297
298 000056 016702 000000G PVNAM: MOV .PSTCN,R2 ; GET NUMBER OF CHARACTERS FOUND
299 000062 022702 000006 CMP #PVNAMX,R2 ; LEGAL PVC CIRCUIT ID?
300 000066 103432 BLO 101$ ; BR IF NO
301 000070 012700 000036' MOV #PVNAM,R0 ; POINT TO STORAGE FOR CIRCUIT ID
302 000074 012701 000006 MOV #PVNAMX,R1 ; GET LENGTH OF CIRCUIT ID
303 000100 112720 000040 10$: MOVB #SPACE,(R0)+ ; INITIALIZE NAME TO SPACES
304 000104 005301 DEC R1 ; MORE TO INITIALIZE?
305 000106 003374 BGT 10$ ; BR IF YES
306 000110 016700 000000G MOV .PSTPT,R0 ; POINT TO START OF CIRCUIT ID
307 000114 012701 000036' MOV #PVNAM,R1 ; POINT TO STORAGE AREA FOR CIRCUIT ID
308 000120 112021 20$: MOVB (R0)+,(R1)+ ; STORE CIRCUIT ID
309 000122 005302 DEC R2 ; MORE TO STORE?
310 000124 003375 BGT 20$ ; BR IF YES
311 000126 016767 000000G 000014' MOV $X3DBS,PVCPSZ ; SET UP DEFAULT MAXIMUM BLOCK SIZE
312 000134 016767 000000G 000016' MOV $X3DWS,PVCWSZ ; SET UP DEFAULT WINDOW SIZE
313 000142 005067 000020' CLR PVCOWN ; INITIALIZE PROCESS OWNER
314 000146 105067 000035' CLRB PVCFLG ; INITIALIZE FLAGS BYTE
315 000152 RETURN
316
317 ; ERRORS
318
319 000154 101$: MSG$R NO ; ILLEGAL CIRCUIT ID
320
321 ; LOGICAL CHANNEL NUMBER
322
323
324 000162 005767 000000G PVLCN: TST .PNUMH ; LEGAL VALUE?
325 000166 001011 BNE 101$ ; BR IF NO
326 000170 016700 000000G MOV .PNUMB,R0 ; GET SPECIFIED LOGICAL CHANNEL NUMBER
327 000174 003406 BLE 101$ ; BR IF ILLEGAL VALUE
328 000176 020027 000000G CMP R0,#CHNLMX ; LEGAL VALUE?
329 000202 101003 BHI 101$ ; BR IF NO
330 000204 010067 000010' MOV R0,PVCLCN ; STORE LOGICAL CHANNEL NUMBER
331 000210 RETURN
332
333 ; ERRORS
334
335 000212 101$: MSG$R 09 ; ILLEGAL LOGICAL CHANNEL NUMBER
336
337 ; COUNTER TIMER
338
339
340 000220 005767 000000G PVCT: TST .PNUMH ; LEGAL VALUE?
341 000224 001004 BNE 101$ ; BR IF NO
342 000226 016767 000000G 000012' MOV .PNUMB,PVCCTM ; SAVE COUNTER TIMER VALUE
343 000234 RETURN
344
345 ; ERRORS
346
347 000236 101$: MSG$R 05 ; ILLEGAL COUNTER TIMER VALUE

```

```

816 002236 110162 000036 40$: MOV B R1,T$SVC+2(R2) ;; STORE PORT NUMBER
817 002242 152762 000200 000044 B1SB #PX$DLM,T$FLAG(R2) ;; INDICATE DLM PVC
818 002250 016762 000000' 000030 MOV HTMR,T$T5(R2) ;; STORE HELLO TIMER
819 002256 016762 000002' 000032 MOV LTIMR,T$T6(R2) ;; AND LISTEN TIMER
820 002264 50$: RESMAP ;; RESTORE PREVIOUS MAPPING
821 002270 60$: RETURN
822
823 000001 .END

```

.TITLE CFGSLT - SCAN CONFIGURATION FILE FOR SLT\$DF  
.IDENT /V05.00/

COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NLT - CONFIG FILE SCAN FOR SLT\$DF FOR SET LINE PROCESSING

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 05-AUG-81  
DECNET-11M/S V3.1  
DECNET-11MPLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RX V1.0

.TITLE CFGSTA - CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 04-JUN-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 3.01 10-JAN-83  
SUPPORT CHANGED CETAB PARAMETERS
- 4.00 07-NOV-83  
DECNET-11M v4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S v4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

CFGSTA      CREATED BY    MACRO    ON 29-JUN-85 AT 00:06      PAGE 4      I 12

MACRO CROSS REFERENCE      CREF    04.00

MACRO NAME      REFERENCES

|         |        |        |        |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| CALL    | 8-102  |        |        |        |        |        |        |        |        |        |
| DBGTP\$ | #9-114 | #9-132 | #9-142 | #9-149 | #9-159 | #9-165 | #9-168 | #9-169 |        |        |
| EMSG\$R | #5-61  | 8-110  | 10-192 | 10-193 | 10-214 | 10-227 | 11-243 |        |        |        |
| GCL\$   | #5-61  |        |        |        |        |        |        |        |        |        |
| ISTAT\$ | #5-61  | 9-114  |        |        |        |        |        |        |        |        |
| MTRAN\$ | #9-114 |        |        |        |        |        |        |        |        |        |
| NTLR\$  | #5-61  | 7-75   | 7-76   | 7-77   | 7-78   | 7-79   |        |        |        |        |
| RETURN  | 8-106  | 10-191 | 10-212 | 10-225 | 11-241 |        |        |        |        |        |
| SLTDF\$ | #5-61  | 5-63   |        |        |        |        |        |        |        |        |
| STAT\$  | #5-61  | 9-118  | #9-121 | #9-124 | #9-127 | #9-130 | #9-134 | #9-137 | #9-140 | #9-144 |
|         | #9-147 | #9-151 | #9-157 | #9-163 | #9-166 | #9-170 | #9-173 |        |        |        |
| TRAN\$  | #5-61  | #9-119 | #9-122 | #9-125 | #9-128 | #9-131 | #9-132 | #9-135 | #9-138 | #9-141 |
|         | #9-142 | #9-145 | #9-148 | #9-149 | #9-152 | #9-158 | #9-159 | #9-164 | #9-165 | #9-167 |
|         | #9-168 | #9-169 | #9-171 |        |        |        |        |        |        |        |



```

414                                     ; OWNER
415
416
417 000550 022767 000003 000000G $VOWNER: CMP #3, PSTCN          ; VALID OWNER?
418 000556 001162                                     BNE 101$          ; BR IF NO
419 000560 016700 000000G                                     MOV , PSTPT, R0    ; POINT TO OWNER
420 000564 005001                                     CLR R1           ;
421 000566                                     CALL $CAT5        ; CONVERT TO RAD50
422 000572 020127 114224                                     CMP R1, #*RXPT   ; IS OWNER XPT?
423 000576 001152                                     BNE 101$          ; BR IF NO - ERROR
424 000600                                     CALL FNDOWN       ; LOOK FOR XPT
425 000604 103547                                     BCS 101$          ; BR IF XPT NOT FOUND
426 000606 016767 000000G 000016'  MOV $SLTA, SLTA      ; MAKE LOCAL COPY OF $SLTA
427 000614 016700 000014'  MOV STASVC, R0      ; GET STATION NUMBER
428 000620 006300                                     ASL R0           ; MAKE IT A WORD INDEX
429 000622 016067 000000G 000002'  MOV $SHTIM(R0), HTIMR ; STORE HELLO TIMER
430 000630 016067 000000G 000004'  MOV $SLTIM(R0), LTIMR ; STORE LISTEN TIMER
431 000636 017700 000000G                                     MOV @PSIPT, R0    ; GET PSI HOME BLOCK ADDRESS
432 000642 001533                                     BEQ 102$          ; NO PSI HOME BLOCK???
433
434 000644                                     SWSTK$ 50$       ; ENTER SYSTEM STATE
435 000650 010004                                     MOV R0, R4       ; SAVE PSI HOME BLOCK ADDRESS
436 000652 012701 000032  MOV #S$LEN, R1      ; GET LENGTH OF SVC DESCRIPTOR BLOCK
437 000656                                     CALL $XALOC       ; ALLOCATE SVC DESCRIPTOR BLOCK
438 000662 103006                                     BCC 10$          ; BR IF SUCCESS
439 000664                                     RETC R0          ; ELSE SET USER C-BIT
440 000676 000507                                     BR 40$          ; AND EXIT
441
442 000700                                     10$: SAVMAP      ; SAVE CURRENT MAPPING
443 000704 010005                                     MOV R0, R5       ; SAVE CURRENT MAPPING
444 000706 010046                                     MOV R0, -(SP)    ; SET UP FOR CONVERSION
445 000710                                     CALL $CEACX       ; CONVERT TO MAPPED ADDRESS
446 000714 012600                                     MOV (SP)+, R0    ; RETRIEVE MAPPED ADDRESS
447 000716 010046                                     MOV R0, -(SP)    ; SAVE VIRTUAL ADDRESS OF BLOCK
448 000720 017746 000000G  MOV @K$AR5, -(SP) ; AND APR BIAS (FOR PLB STORAGE)
449 000724 010002                                     MOV R0, R2       ; COPY MAPPED ADDRESS
450 000726 105022 15$: CLR R2          ; INITIALIZE BLOCK
451 000730 005301                                     DEC R1           ; MORE TO INITIALIZE?
452 000732 003375                                     BGT 15$          ; BR IF YES
453 000734 116760 000021' 000002  MOVB CTL$SVC, S$CTL(R0) ; STORE CONTROLLER NUMBER
454 000742 116760 000014' 000003  MOVB STASVC, S$STA(R0) ; STORE STATION NUMBER
455 000750 016760 000006' 000004  MOV PKT$SVC, S$PKSZ(R0) ; STORE PACKET SIZE
456 000756 016760 000010' 000010  MOV FLGSVC, S$FLG(R0)  ; STORE FLAGS BYTE
457 000764 116760 000023' 000013  MOVB RCL$SVC, S$RCL(R0) ; STORE RECALL TIMER
458 000772 116760 000022' 000006  MOVB WND$SVC, S$WIND(R0) ; STORE WINDOW SIZE
459 001000 016760 000012' 000016  MOVB RTSVC, S$TMR(R0)  ; STORE RETRY TIMER
460 001006 116760 000024' 000007  MOVB COUNT, S$ADSZ(R0) ; STORE SIZE OF DTE ADDRESS (DIGITS)
461 001014 012702 000010  MOV #<DTELEN+1>/2, R2 ; GET NUMBER OF BYTES IN DTE ADDRESS
462 001020 010001                                     MOV R0, R1       ; GET BLOCK ADDRESS
463 001022 062701 000020  ADD #S$DTE, R1    ; POINT TO DTE ADDRESS
464 001026 012703 000044'  MOV #DTEPCK, R3   ; GET DTE ADDRESS
465 001032 112321 17$: MOVB (R3)+, (R1)+ ; STORE DTE ADDRESS
466 001034 005302                                     DEC R2           ; MORE TO STORE?
467 001036 003375                                     BGT 17$          ; BR IF YES
468 001040 116760 000020' 000030  MOVB OWNSVC, S$OWNER(R0) ; STORE OWNER
469 001046 062704 000036  ADD #H$SVC, R4     ; POINT TO SVC LISTHEAD
470 001052 010401 20$: MOV R4, R1          ; SAVE ADDRESS OF PREVIOUS BLOCK

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES   |
|---------|------------|--|
| \$ERR3M | 000340 R   | #7-146 12-378  |
| \$ERR3Z | 000374 R   | #7-147 13-492  |
| \$EXIT  | = 000000   | #9-185   |
| \$FAIL  | = 177777   | #9-185   |
| \$GPRM  | = *****    | 9-185  |
| \$HEADR | = ***** GX | 13-439 14-515  |
| \$LAMDA | = 000000   | #9-185   |
| \$LINKX | = ***** GX | 13-479   |
| \$NUMBR | = 000002   | #9-185   |
| \$QSVC  | 000000 RG  | #8-165   |
| \$RAD50 | = 000016   | #9-185   |
| \$RONLY | = *****    | 9-185 9-185  |
| \$SAVRG | = ***** GX | 12-407   |
| \$SLTA  | = ***** GX | 10-280 13-426  |
| \$STRNG | = 000004   | #9-185   |
| \$SUBXP | = 000010   | #9-185   |
| \$TRIB  | = ***** GX | 10-294   |
| \$XALOC | = ***** GX | 13-437   |
| \$HITIM | = ***** GX | 13-429   |
| \$SLTIM | = ***** GX | 13-430   |
| \$SFLG  | = 177777   | #9-185   |
| \$SKEY  | = 177777   | #9-185   |
| .PCHAR  | = ***** GX | 17-637   |
| .PNUMB  | = ***** GX | 10-277 10-279 10-281 10-291 10-293 10-294 11-310 11-322 11-324 |
|         |            | 11-336 11-348 11-350 12-362                                    |
| .PNUMH  | = ***** GX | 10-275 10-289 11-308 11-320 11-334 11-346 12-360               |
| .PSTCN  | = ***** GX | 10-261 13-417  |
| .PSTPT  | = ***** GX | 10-263 13-419  |
| .TPARS  | = ***** GX | 8-171  |

```

278
279
280
281
282
283
284 000400
285 000404 016746 000000G
286 000410 042716 177770
287 000414 022726 000001
288 000420 001065
289 000422 126100 000013
290 000426 001003
291 000430 016746 000000G
292 000434 000404
293 000436 006300
294 000440 006300
295 000442 016046 000004G
296 000446 000316
297 000450 042716 177760
298 000454 011602
299 000456 006316
300 000460 062716 000000'
301 000464 013600
302 000466 016701 000000G
303 000472 126161 000003 000002
304 000500 001425
305 000504 012701 000010
306 000506 032767 000004 000000G
307 000514 001402
308 000516 012701 000012
309 000522
310 000526 010001
311 000530 017700 000000G
312 000534
313 000540 062700 000002
314 000544 042700 177400
315 000550 000300
316 000552 000411
317
318
319
320 000554 012700 000002
321 000560 020227 000017
322 000564 001001
323 000566 006300
324 000570 000300
325 000572 000401
326
327
328
329 000574 005000
330
331 000576 105067 000001G
332 000602 050067 000000G
333 000606
334 000612

; SETTMO - SET UP TIMEOUT VALUE
; INPUTS:
; R0 - UNIT NUMBER
SETTMO: SAVRG <R0,R1> ; SAVE SOME REGISTERS
MOV .PNUMB,-(SP) ; GET CHARACTERISTICS WORD 1
BIC #^C<7>,(SP) ; ISOLATE PROTOCOL BITS
CMP #C1.DCP,(SP)+ ; IS THIS A DDCMP LINE?
BNE 40$ ; BR IF NO
L.UNT(R1),R0 ; IS THIS THE KEY UNIT?
BNE 5$ ; BR IF NO
MOV $$DCHA,-(SP) ; GET CHARACTERISTICS WORD 0
BR 7$
5$: ASL R0 ; CONVERT TO DOUBLE WORD INDEX
ASL R0
MOV $$DCHA+4(R0),-(SP) ; GET CHARACTERISTICS WORD 0
7$: SWAB (SP) ; GET HIGH BYTE
BIC #^C<17>,(SP) ; ISOLATE SPEED BITS
MOV (SP),R2 ; SAVE THEM
ASL (SP) ; CONVERT TO WORD INDEX
ADD #SPEEDT,(SP) ; INDEX INTO TRANSLATION TABLE
MOV @R2,R0 ; GET CORRECT SPEED
MOV $SLTA,R1 ; GET SLT ADDRESS
CMPB L.DLC(R1),L.DDM(R1) ; IS THIS A COMBINED DDM/DLC?
BEQ 20$ ; BR IF YES
MOV #8.,R1 ; ASSUME 8 BITS
BIT #CH.SYN,$$DCHA ; IS TYPE SYNCH?
BEQ 10$ ; BR IF YES
MOV #10.,R1 ; ELSE MUST BE ASYNCH
CALL $DIV ; CALCULATE SPEED/# BITS
MOV R0,R1 ; GET RESULT
MOV @RDBSZ,R0 ; GET SIZE OF RDB'S
CALL $DIV ; CALCULATE TIME OUT VALUE
ADD #2,R0 ; ADD FUDGE FACTOR
BIC #^C<377>,R0 ; ISOLATE TIMEOUT VALUE
SWAB R0 ; GET IN HIGH BYTE
BR 50$ ; AND FINISH UP

; DDM = DLC
20$: MOV #2,R0 ; GET NUMBER OF BUFFERS (ASSUME SPEED <=9600)
CMP R2,#17 ; EXTERNAL CLOCK?
BNE 30$ ; BR IF NO
ASL R0 ; ELSE NUMBER OF BUFFERS = 4
30$: SWAB R0 ; GET INTO HIGH BYTE
BR 50$ ; AND STORE NUMBER OF BUFFERS

; NON-DDCMP LINE
40$: CLR R0 ;
50$: CLRB .PNUMB+1 ; STORE TIMEOUT VALUE (OR NUMBER OF BUFFERS)
BIS R0,.PNUMB ;
RESRG <R1,R0> ; RESTORE REGISTERS
RETURN

```

## MACRO DEFINITIONS

```
55 .SBTTL MACRO DEFINITIONS
56
57 ;
58 ; LIBRARY MACROS
59 ;
60 .MCALL ISTAT$,STATE$,TRAN$,EMSG$R,NTLR$,SLTDF$,PDVDF$
61
62 PDVDF$ ; DEFINE PDV OFFSETS AND SYMBOLS
63 SLTDF$ ; DEFINE SLT OFFSETS AND SYMBOLS
64
65 ;
66 ; LOCAL MACRO DEFINITIONS
67 ;
68 .MACRO REJ$ ; RETURN TO CALLER+2
69 ADD #2,(SP) ; INDICATE NO SYNTAX ERROR
70 CLR SYNERR
71 .ENDM
```

ALCMEM - ALLOCATE A BLOCK OF ME MACRO V05.03b Saturday 29-Jun-85 <sup>J</sup>00:01<sup>1</sup>  
Table of contents

6- 123 \$AMEM - ALLOCATE MEMORY IN NTPool  
9- 229 \$DMEM - DEALLOCATE MEMORY FROM PARTITION

.TITLE CEALL - ALLOCATE BYTES OF POOL  
.IDENT /V05.00/

COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

GENERAL ALLOCATION/DE-ALLOCATION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 27-FEB-78  
VERSION 2.0 RELEASE
- 2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

91                                     .SBTTL $QCNT - LOOK FOR CNT$DF MACRO
92
93
94                                     ;+
95                                     ;$QCNT - LOOK FOR CNT$DF MACRO
96                                     ;
97                                     ; INPUTS:
98                                     ;     NONE
99                                     ;
100                                    ; OUTPUTS:
101                                    ;     C-BIT=SUCCESS/FAILURE
102                                    ;     R3,R4,R5=DESTROYED
103                                    ; -
104 000000 012705 000000' $QCNT:: MOV #CNTDF,R5 ; STATE TABLE ADDRESS
105 000004 005001          CLR R1 ; FULL KEYWORD MATCH LENGTH
106 000006 012702 000000' MOV #CNTKW,R2 ; KEYWORD TABLE ADDRESS
107 000012 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
108 000016 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
109 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
110 000026          CALL TPARS ; GO DO THE PARSE
111 000032 103003          BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
112 000034 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
113 000040 001401          BEQ 101$ ; IF EQ, YES
114 000042          20$: RETURN
115
116                                     ;
117                                     ; ERROR CONDITION
118
119 000044          101$: MSG$R 1T ; SYNTAX ERROR

```

```

90          .SBTTL  TPARS STATE TABLES
91
92 000052      ISTAT$  DDMST,DDMKW
93
94          : DDM DEFINITION (DDM$DF)
95          :
96 000052      STATES$  DDMDF
97 000052      TRANS$   %DDM$DF%,DDMNAM,,1,SYNERR
98 000052      TRANS$   %DLC$DF%,,,1,SYNERR
99 000052      STATES$  DDMNAM          ; DDM NAME
100
101          .IF DF  M$$MGE
102
103 000052      TRANS$   $RAD50,,D.NAME
104 000052      STATES$
105 000052      TRANS$   <','>
106 000052      STATES$          ; FLAG BITS
107 000052      TRANS$   !BITS
108 000052      STATES$          ; PRIORITY
109 000052      TRANS$   $NUMBR
110 000052      STATES$
111 000052      TRANS$   !END,$EXIT
112 000052      TRANS$   <','>
113 000052      STATES$          ; LINE TABLE COUNT
114 000052      TRANS$   $NUMBR,,P.INC
115 000052      STATES$
116 000052      TRANS$   !END,$EXIT
117 000052      TRANS$   <','>
118 000052      STATES$          ; CONTROLLER TABLE COUNT
119 000052      TRANS$   $NUMBR,$EXIT,D.EXT
120
121          .IFF
122
123          TRANS$   $RAD50,$EXIT,D.NAME
124
125          .ENDC
126
127          : FLAG BIT DEFINITIONS PLUS TRAILING COMMA
128          :
129 000052      STATES$  BITS
130 000052      TRANS$   %LF.X2P%,BITS1,,FG.X2P,CFGFLG
131 000052      TRANS$   %ZF.X3P%,BITS1,,FG.X3P,CFGFLG
132 000052      TRANS$   $RAD50,BITS1
133 000052      TRANS$   <','>,$EXIT
134 000052      STATES$  BITS1
135 000052      TRANS$   <','>,BITS2
136 000052      TRANS$   '+,BITS2
137 000052      TRANS$   <','>,$EXIT
138 000052      STATES$  BITS2
139 000052      TRANS$   %LF.X2P%,BITS1,,FG.X2P,CFGFLG
140 000052      TRANS$   %ZF.X3P%,BITS1,,FG.X3P,CFGFLG
141 000052      TRANS$   $RAD50,BITS1
142 000052
143          : CHECK FOR END OF SOURCE LINE
144          :
145 000052      STATES$  END
146

```



```

140 .SBTTL LOOK FOR DTE$DF MACRO
141
142 ;+
143 ;$QDTE - LOOK FOR DTE$DF MACRO
144 ;
145 ; INPUTS:
146 ; NONE
147 ;
148 ; OUTPUTS:
149 ; C-BIT=SUCCESS/FAILURE
150 ; R3,R4,R5=DESTROYED
151 ;
152 $QDTE:: MOV #DTE$DF,R5 ; STATE TABLE ADDRESS
153 CLR R1 ; FULL KEYWORD MATCH LENGTH
154 MOV #DTE$KW,R2 ; KEYWORD TABLE ADDRESS
155 MOV C$G$Z,R3 ; RECORD LENGTH
156 MOV #C$G$B,R4 ; RECORD BUFFER ADDRESS
157 CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
158 CALL TPARS ; GO DO THE PARSE
159 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
160 DEC SYNERR ; DID SYNTAX ERROR OCCUR?
161 BEQ 101$ ; IF EQ, YES
162 20$: RETURN
163
164 ; ERROR CONDITION
165
166 101$: MSG$R 1T ; SYNTAX ERROR
167 000044
  
```

CFGDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:02 Page 20-2  
Symbol table

\$QDTE 000000RG      \$\$SUBXP= 000010      \$\$\$R = 000010      .PCHAR= \*\*\*\*\* GX      .PSTCN= \*\*\*\*\* GX  
\$RAD50= 000016      \$\$FLG= 177777      \$\$\$STA= 000000      .PNUMB= \*\*\*\*\* GX      .PSTPT= \*\*\*\*\* GX  
\$SAVRG= \*\*\*\*\* GX      \$\$KEY= 000002      \$\$\$TMP= 000012R      005 .PNUMH= \*\*\*\*\* GX      .TPARS= \*\*\*\*\* GX  
\$STRNG= 000004

. ABS. 177776    000 (RW,I,GBL,ABS,OVR)  
      000740    001 (RW,I,LCL,REL,CON)  
DATA 000244    002 (RW,D,LCL,REL,CON)  
\$STATE 000154    003 (RW,D,LCL,REL,CON)  
\$KTAB 000006    004 (RW,D,LCL,REL,CON)  
\$KSTR 000016    005 (RW,D,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 16635 Words ( 65 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:44.10  
SY:CFGDTE.V2,[132,134]CFGDTE/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFGDTE

CFGFEA CREATED BY MACRO ON 29-JUN-85 AT 00:03 PAGE 1 J 7  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL    | VALUE      | REFERENCES          |
|-----------|------------|---------------------|
| CFGFBF    | = ***** GX | 6-90                |
| CFGSZ     | = ***** GX | 6-89                |
| FEADF     | 000000 R   | 6-86                |
| FEAKW     | 000000 RG  | 6-88 #7-106         |
| FEAST     | 000000 RG  | #7-106              |
| F.CH1     | 000052 R   | #8-144              |
| F.CH2     | 000074 R   | #8-151              |
| F.FEA     | 000116 R   | #8-158              |
| F.MOD     | 000126 R   | #8-163              |
| F.SET     | 000136 R   | #8-168              |
| SYNERR    | = ***** GX | *6-91 *8-146 *8-153 |
| \$ALPHA   | = 000022   | #7-106              |
| \$ANY     | = 000020   | #7-106              |
| \$BLANK   | = 000006   | #7-106              |
| \$DIGIT   | = 000024   | #7-106              |
| \$DNUMB   | = 000014   | #7-106              |
| \$EOS     | = 000012   | #7-106              |
| \$ERRIT   | = ***** GX | 6-100               |
| \$EXIT    | = 000000   | #7-106              |
| \$FAIL    | = 177777   | #7-106              |
| \$FEATR   | = ***** GX | *8-158 *8-168       |
| \$GPRM    | = *****    | 7-106               |
| \$LAMDA   | = 000000   | #7-106              |
| \$NUMBR   | = 000002   | #7-106              |
| \$QFEA    | 000000 RG  | #6-86               |
| \$RAD50   | = 000016   | #7-106              |
| \$RONLY   | = *****    | 7-106 7-106         |
| \$STRNG   | = 000004   | #7-106              |
| \$SUBXP   | = 000010   | #7-106              |
| \$SDEV    | = ***** GX | 8-144 8-151         |
| \$\$\$FLG | = 177777   | #7-106              |
| \$\$\$KEY | = 177777   | #7-106              |
| .PCHAR    | = ***** GX | 8-144 8-151         |
| .PNUMB    | = ***** GX | 8-158 8-163 8-168   |
| .TPARS    | = ***** GX | 6-92                |

```

349
350
351      ; MAXIMUM BLOCK SIZE
352
353      PVPSZ: CALL    CHKBLK      ; CHECK FOR LEGAL BLOCK SIZE VALUE
354             BCS     101$      ; BR IF ERROR
355             MOV     RO,PVCPSZ  ; SAVE MAXIMUM BLOCK SIZE
356             RETURN
357
358      ; ERRORS
359
360      101$: MSG$R P8           ; ILLEGAL MAXIMUM BLOCK SIZE
361
362      ; WINDOW SIZE
363
364      PVWSZ: CALL    CHKWND      ; CHECK FOR LEGAL WINDOW SIZE VALUE
365             BCS     101$      ; BR IF ILLEGAL VALUE
366             MOV     RO,PVCWSZ  ; SAVE WINDOW SIZE
367             RETURN
368
369      ; ERRORS
370
371      101$: MSG$R 00           ; ILLEGAL WINDOW SIZE
372
373      ; PROCESS OWNER
374
375      PVOWN: CMP     #3,.PSTCN   ; LEGAL PROCESS OWNER?
376             BLO     101$      ; BR IF NO
377             MOV     .PSTPT,RO  ; POINT TO STRING TO CONVERT TO RAD50
378             CLR     R1         ; PERIOD IS NOT A VALID TERMINATOR
379             CALL    $CAT5      ; CONVERT TO RAD50
380             MOV     RO,PVCOWN  ; SAVE PROCESS OWNER NAME
381             RETURN
382
383      ; ERRORS
384
385      101$: MSG$R N1           ; ILLEGAL PROCESS OWNER NAME
386
387      ; FLAGS BYTE
388
389      PVFG: TST     .PNUMH      ; LEGAL VALUE?
390             BNE     101$      ; BR IF NO
391             TSTB    .PNUMB+1   ; LEGAL VALUE?
392             BNE     101$      ; BR IF NO
393             MOV     .PNUMB,PVCFLG ; SAVE FLAGS BYTE
394             RETURN
395
396      ; ERRORS
397
398      101$: MSG$R N2           ; ILLEGAL FLAGS BYTE VALUE
399
400
401

```

|                  |                   |                  |                 |                    |
|------------------|-------------------|------------------|-----------------|--------------------|
| ALLXCB 000724R   | E\$NRTP 000005    | J\$SRDN= 000000  | L\$AUC 000042   | L.NMST 000020      |
| A\$CHK= 000000   | E\$NSEG 000010    | KSARS = ***** GX | L\$CHLS 000034  | L.NSTA 000014      |
| A\$CPS= 000000   | E\$NTIM 000046    | K\$CNT= 177546   | L\$CHTB 000030  | L.OWNR 000021      |
| A\$PRI= 000000   | E\$NUSE 000004    | K\$CSR= 177546   | L\$CLEN= 000044 | L.UNT 000013       |
| A\$TRP= 000000   | E\$STRT 000006    | K\$LDC= 000000   | L\$CTEN 000032  | M\$CRB= 000124     |
| BIASX 001610R    | E\$XPR= 000000    | K\$TPS= 000074   | L\$CTIM 000040  | M\$CRX= 000000     |
| CERR = ***** GX  | FLAG 000032R 002  | LD\$LP = 000000  | L\$DTEA 000020  | M\$FCR= 000000     |
| CFGBF = ***** GX | FM18 = ***** GX   | LF.ACT= 100000   | L\$DTEL 000017  | M\$MGE= 000000     |
| CFGSZ = ***** GX | FM18B = ***** GX  | LF.BRO= 000400   | L\$GLEN 000134  | M\$NET= 000000     |
| CFLIN = ***** GX | FM.8 = 000000     | LF.BWT= 000007   | L\$LEN 000122   | M\$OVR= 000000     |
| CF\$BLK= 000102  | FM.8B = 000000    | LF.ENA= 002000   | L\$LLCH 000016  | N\$ADJ1 000072     |
| CHKBLK= ***** GX | FNDHSH 001452R    | LF.LPB= 001000   | L\$LNK 000000   | N\$ADJ2 000074     |
| CHKDLM 000164R   | FNDPDV= ***** GX  | LF.MDC= 000100   | L\$MCHN 000036  | N\$CACH 000062     |
| CHKPVC 001050R   | FNDPRT 001172R    | LF.MFL= 004000   | L\$NETW 000114  | N\$CRC 000120      |
| CHKWND= ***** GX | F\$SLVL= 000001   | LF.MTP= 000020   | L\$NIRE 000072  | N\$HC1 000052      |
| CHNLMX= ***** GX | G\$STPP= 000000   | LF.PAC= 000200   | L\$NLRE 000104  | N\$HC2 000056      |
| CHNLNO 000014R   | G\$STSS= 000000   | LF.RDY= 040000   | L\$NMMA 000076  | N\$LV1 000002      |
| C\$DTE 000012    | G\$STTK= 000000   | LF.REA= 010000   | L\$NMAS 000074  | N\$LV2 000010      |
| C\$FLG 000015    | G\$SWRD= 000000   | LF.SER= 000040   | L\$NMST 000003  | N\$MHC1 000036     |
| C\$GNAM 000016   | HF\$DLM= 000002   | LF.TJM= 000010   | L\$NNRE 000106  | N\$MHC2 000044     |
| C\$LCN 000010    | HF\$GWY= 000010   | LF.UNL= 020000   | L\$NNRY 000046  | N\$PLD 000016      |
| C\$LEN 000015    | HF\$HOS= 000004   | LF.X2P= 000000   | L\$NRCA 000066  | N\$PRI 000076      |
| C\$LNK 000000    | HF\$XDF= 000020   | LLCTA = ***** GX | L\$NRFS 000100  | N\$ROA1 000022     |
| C\$NAM 000002    | HSHADD= ***** GX  | LNKEND= ***** GX | L\$NRPK 000056  | N\$ROA2 000030     |
| C\$NML 000002    | HSHSZ = ***** GX  | LN\$OFF= 000001  | L\$NRRE 000105  | N\$RTMX 000014     |
| C\$PORT 000014   | HTIMR 000000R 002 | LN\$ON = 000000  | L\$NRST 000107  | N\$RTM1 000014     |
| C\$CKP= 000000   | H\$CUG 000010     | LN\$SHU= 000002  | L\$NTBY 000052  | N\$RTM2 000015     |
| C\$SORE= 000400  | H\$DST 000012     | LN.CLO= 000000   | L\$NTCA 000070  | N\$RT1 000000      |
| C\$SRSH= 177564  | H\$D29 000014     | LN.DUM= 000005   | L\$NTFS 000102  | N\$RT2 000006      |
| DLMCTL 000044R   | H\$FLG 000000     | LN.LOA= 000004   | L\$NTPK 000062  | N\$TCTL 000112     |
| DLMIND 000046R   | H\$GLEN 000104    | LN.LOO= 000003   | L\$OMST 000002  | N\$TLC 000100      |
| DLMPDV 000022R   | H\$GLT 000044     | LN.OAU= 000003   | L\$PCS 000112   | N\$TNC 000114      |
| DLMSLN 000030R   | H\$GNAM 000050    | LN.OFF= 000001   | L\$PLS 000110   | N\$TRC 000106      |
| DLMSLT 000024R   | H\$GNML= 000020   | LN.ON = 000000   | L\$QCNT 000122  | N\$TTCB 000110     |
| DLMSTA 000026R   | H\$GPT 000046     | LN.OOP= 000004   | L\$QUEH 000124  | N\$VER 000066      |
| DLMTLC 002034R   | H\$HITS 000034    | LN.OPE= 000001   | L\$QUET 000130  | N\$XLEN 000124     |
| DLMUNT 000045R   | H\$HLEN 000044    | LN.REF= 000002   | L\$RTRY 000007  | N\$XACC= 000001    |
| DTEDES= ***** GX | H\$LBDA 000070    | LN.SER= 000002   | L\$SLN 000004   | N\$XBUF= 000001    |
| D\$BUG= 177514   | H\$LBDN 000072    | LN.STA= 000017   | L\$ST 000005    | N\$XLDV= 000001    |
| D\$ISK= 000000   | H\$LDTE 000002    | LN.SUB= 000360   | L\$TCLZ 000044  | N\$XMC= 000001     |
| D\$SL11= 000001  | H\$LEN 000042     | LN.TRI= 000006   | L\$TIM 000006   | N\$XMLL= 000001    |
| D\$SYNC= 000000  | H\$LOTS 000032    | LP\$EJP= 002000  | L\$SASG= 000000 | N\$XMOV= 000010    |
| D\$SYNM= 000000  | H\$NETW 000024    | LP\$ENB= 004000  | L\$SDRV= 000000 | N\$XMT= 000001     |
| END 000154R      | H\$NML = 000006   | LP\$PCT= 001400  | L\$SP11= 000001 | N\$XPEM= 000001    |
| E\$NBR 000014    | H\$NPT 000022     | LP\$TMR= 100000  | L\$S11R= 000000 | OWNER 000110R 003  |
| E\$NBS 000020    | H\$PTB 000020     | LP\$UP = 010000  | L.COST 000015   | PCVKW 000000RG 004 |
| E\$NCR 000034    | H\$PVC 000006     | LP\$WTD= 020000  | L.CTL 000012    | PDVID = ***** GX   |
| E\$NCS 000036    | H\$RDE 000004     | LP\$WTS= 040000  | L.CVA 177776    | PDVNM = ***** GX   |
| E\$NIC 000044    | H\$RNW 000042     | L\$GDN= 000004   | L.DDM 000002    | PDVTA = ***** GX   |
| E\$NLEN 000050   | H\$SVC 000036     | L\$OFF= 000000   | L.DDS 000004    | PFLAGS 000132R 003 |
| E\$NLLA 000012   | H\$TRB 000016     | L\$STR= 000002   | L.DLC 000003    | PF\$BLK= 020000    |
| E\$NLNK 000000   | H\$XAVL 000100    | L\$SUP = 000003  | L.DLM 000006    | PF\$CLC= 010000    |
| E\$NML 000040    | H\$XBIA 000074    | L\$SWT = 000001  | L.DLS 000010    | PF\$DLM= 100000    |
| E\$NMR 000024    | H\$X29C 000040    | LTIMR 000002R    | L.FLG 000000    | PF\$EIP= 000002    |
| E\$NMS 000030    | INIXCB 001266R    | L\$ACHN 000012   | L.KRBA 000016   | PF\$ENB= 000001    |
| E\$NNOD 000002   | INVCIR 001720R    | L\$APVC 000014   | L.LEN = 000022  | PF\$FAI= 004000    |
| E\$VRT 000042    | J\$RAR= 000000    | L\$ASVC 000010   | L.MPF = 000022  | PF\$FM1= 000100    |

```
52          .SBTTL  MACRO DEFINITIONS
53
54          :
55          : LIBRARY MACROS
56          :
57          .MCALL  MSG$R, ISTAT$, STATE$, TRANS$, SLTDF$
58          SLTDF$                                ; DEFINE SLT OFFSETS
59
60          :
61          : LOCAL MACROS
62          :
63          :
64          : REJECT TPARS TRANSITION
65          :
66          .MACRO  REJ$
67          ADD     #2, (SP)                        ; RETURN TO CALLER+2
68          CLR     SYNERR                          ; INDICATE NO SYNTAX ERROR
69          .ENDM   REJ$
```

```
57 .SBTTL MACRO DEFINITIONS
58
59 : LIBRARY MACROS
60 :
61 .MCALL EMSG$,GCL$,ISTAT$,NTLR$,STAT$,TRAN$,SLTDF$
62
63 000000 SLTDF$ ; DEFINE SLT OFFSETS
```

\*\*FILE\*\*ID\*\*CFG SVC

```

CCCCCCCC FFFFFFFF GGGGGGGG SSSSSSSS VV VV CCCCCCCC
CCCCCCCC FFFFFFFF GGGGGGGG SSSSSSSS VV VV CCCCCCCC
CC FF GG SS VV VV CC
CC FF GG SS VV VV CC
CC FF GG SS VV VV CC
CC FFFFFFFF GG SSSSSS VV VV CC
CC FFFFFFFF GG SSSSSS VV VV CC
CC FF GG GGGGGG SS VV VV CC
CC FF GG GGGGGG SS VV VV CC
CC FF GG GG SS VV VV CC
CC FF GG GG SS VV VV CC
CCCCCCCC FF GGGGGG SSSSSSSS VV VV CCCCCCCC
CCCCCCCC FF GGGGGG SSSSSSSS VV VV CCCCCCCC

```

```

LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```



|     |        |        |               |          |   |
|-----|--------|--------|---------------|----------|---|
| 471 | 001054 | 011404 | MOV           | (R4),R4  | :: GET ADDRESS OF NEXT BLOCK IN LIST    |
| 472 | 001056 | 001405 | BEQ           | 30\$     | :: BR IF END OF LIST - LINK IT HERE     |
| 473 | 001060 | 010446 | MOV           | R4,-(SP) | :: SET UP UNMAPPED ADDR FOR CONVERSION  |
| 474 | 001062 |        | CALL          | \$CEACX  | :: CONVERT TO MAPPED ADDRESS            |
| 475 | 001066 | 012604 | MOV           | (SP)+,R4 | :: RETRIEVE MAPPED ADDRESS              |
| 476 | 001070 | 000770 | BR            | 20\$     | :: GET NEXT BLOCK IN LIST               |
| 477 | 001072 | 010546 | 30\$: MOV     | R5,-(SP) | :: GET UNMAPPED ADDR OF CURRENT BLOCK   |
| 478 | 001074 | 010146 | MOV           | R1,-(SP) | :: GET ADDRESS OF PREVIOUS BLOCK        |
| 479 | 001076 |        | CALL          | \$LINKX  | :: LINK BLOCK INTO LIST                 |
| 480 | 001102 | 012600 | MOV           | (SP)+,R0 | :: GET APR BIAS OF SVC DESCRIPTOR BLOCK |
| 481 | 001104 | 012601 | MOV           | (SP)+,R1 | :: GET VIRTUAL ADDRESS OF SVC BLOCK     |
| 482 | 001106 |        | CALL          | SETTLC   | :: SET UP TRANSPORT LINE COUNTERS       |
| 483 | 001112 |        | RESMAP        |          | :: RESTORE PREVIOUS MAPPING             |
| 484 | 001116 |        | 40\$: RETURN  |          |   |
| 485 |        |        |               |          |   |
| 486 | 001120 | 103407 | 50\$: BCS     | 103\$    | ; BR IF ERROR                           |
| 487 | 001122 |        | RETURN        |          |   |
| 488 |        |        | ::            |          |   |
| 489 |        |        | :: ERROR      |          |   |
| 490 |        |        | ::            |          |   |
| 491 | 001124 |        | 101\$: MSG\$R | 3J       | ; ILLEGAL OWNER                         |
| 492 | 001132 |        | 102\$: MSG\$R | 3Z       | ; PSI NOT LOADED                        |
| 493 | 001140 |        | 103\$: MSG\$R | 3K       | ; ALLOCATION FAILURE                    |

## MACRO CROSS REFERENCE

CREF 04.00

## MACRO NAME REFERENCES

|         |        |        |        |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| ASL\$   | #5-60  | 16-612 |        |        |        |        |        |        |        |        |
| BIAS    | #5-100 | 15-364 |        |        |        |        |        |        |        |        |
| CALL    | 8-171  | 10-265 | 12-389 | 12-397 | 12-411 | 13-421 | 13-424 | 13-434 | 13-437 | 13-445 |
|         | 13-474 | 13-479 | 13-482 | 14-511 | 14-513 | 17-639 |        |        |        |        |
| CTDRF\$ | #5-59  | 5-62   |        |        |        |        |        |        |        |        |
| DBGTP\$ | #9-185 | #9-212 | #9-219 | #9-244 | #9-248 |        |        |        |        |        |
| EMSG\$R | #5-59  | 8-179  | 10-301 | 11-315 | 11-329 | 11-341 | 11-355 | 12-367 | 12-378 | 12-402 |
|         | 13-491 | 13-492 | 13-493 |        |        |        |        |        |        |        |
| ISTAT\$ | #5-59  | 9-185  |        |        |        |        |        |        |        |        |
| MAP     | #5-94  | 15-561 |        |        |        |        |        |        |        |        |
| MTRAN\$ | #9-185 |        |        |        |        |        |        |        |        |        |
| NTLR\$  | #5-59  | 7-138  | 7-139  | 7-140  | 7-141  | 7-142  | 7-143  | 7-144  | 7-145  | 7-146  |
|         | 7-147  |        |        |        |        |        |        |        |        |        |
| PHBDF\$ | #5-60  | 5-64   |        |        |        |        |        |        |        |        |
| PLBDF\$ | #5-60  | 5-67   |        |        |        |        |        |        |        |        |
| REJ\$   | #5-75  | 10-283 | 10-296 |        |        |        |        |        |        |        |
| RESMAP  | #5-88  | 13-483 | 15-582 |        |        |        |        |        |        |        |
| RET\$   | #5-60  | 13-439 | 14-515 |        |        |        |        |        |        |        |
| RETURN  | 8-175  | 10-270 | 10-284 | 10-297 | 11-311 | 11-325 | 11-337 | 11-351 | 12-363 | 12-374 |
|         | 12-390 | 12-398 | 12-412 | 13-484 | 13-487 | 14-517 | 15-583 | 16-620 | 17-643 |        |
| SAVMAP  | #5-82  | 13-442 | 15-559 |        |        |        |        |        |        |        |
| SLTDF\$ | #5-60  | 5-65   |        |        |        |        |        |        |        |        |
| SOB     | 15-570 |        |        |        |        |        |        |        |        |        |
| STATE\$ | #5-59  | 9-189  | #9-192 | #9-195 | #9-198 | #9-201 | #9-204 | #9-207 | #9-210 | #9-214 |
|         | #9-217 | #9-221 | #9-224 | #9-227 | #9-230 | #9-233 | #9-236 | #9-239 | #9-242 | #9-246 |
|         | #9-250 | #9-253 |        |        |        |        |        |        |        |        |
| SVCDF\$ | #5-60  | 5-66   |        |        |        |        |        |        |        |        |
| SWSTK\$ | 13-434 | 14-511 |        |        |        |        |        |        |        |        |
| TRAN\$  | #5-59  | #9-190 | #9-193 | #9-196 | #9-199 | #9-202 | #9-205 | #9-208 | #9-211 | #9-212 |
|         | #9-215 | #9-218 | #9-219 | #9-222 | #9-225 | #9-228 | #9-231 | #9-234 | #9-237 | #9-240 |
|         | #9-243 | #9-244 | #9-247 | #9-248 | #9-251 |        |        |        |        |        |
| XPDD\$  | #5-59  | 5-63   |        |        |        |        |        |        |        |        |

```

336      ;
337      ; SECONDARY CSR (UNT$DF)
338      ;
339      000614 032767 000000G 000000G U.XCSR: BIT    #FL.LMC,$FLAGS ; COMIOP DEVICE?
340      000622 001447      BEQ    111$      ; IF EQ, NO
341      000624 016701      MOV    UCNT,R1      ; GET UNIT COUNT
342      000630 032767 000000G 000000G BIT    #FL.KMX,$FLAGS ; COMIOP MUX DEVICE?
343      000636 001406      BEQ    10$      ; IF EQ, NO
344      000640 032701 000007      BIT    #7,R1      ; IS THIS THE FIRST, NINTH, ETC.?
345      000644 001036      BNE    111$      ; IF NE, NO
346      000646 006201      ASR    R1      ; CONVERT TO A WORD INDEX
347      000650 006201      ASR    R1      ;
348      000652 000401      BR     20$      ; ...
349      000654 006301      10$: ASL    R1      ; CONVERT TO A WORD INDEX
350      000656 016700 000000G 20$: MOV    ..$CSR,R0      ; ASSUME SECONDARY CSR ALREADY PRESENT
351      000662 016702 000000G      MOV    $SLTA,R2      ; GET THE SLT ADDRESS
352      000666 032767 000000G 000000G BIT    #MS.SEC,$MISS      ; IS THE SECONDARY CSR MISSING ?
353      000674 001412      BEQ    25$      ; IF EQ, NO .. ALREADY TESTED FOR ONLINE
354      000676      23$: CALL  GETCSR      ; GET THE CSR VALUE
355      000702 103414      BCS    101$      ; .. ILLEGAL NUMERIC VALUE
356      000704 126762 000000G 000013      UCNT,L.UNIT(R2) ; IS THIS THE KEY UNIT ?
357      000712 001003      BNE    25$      ; NO
358      000714      CALL  $NLC$CSR      ; TEST FOR ONLINE
359      000720 103413      BCS    121$      ; OFFLINE
360      000722 010061 000000G 25$: MOV    R0,$$CSR(R1)      ; STORE SECONDARY CSR VALUE
361      000726 005267 000000G      INC    UCNT      ; INCREMENT UNIT COUNT
362      000732      RETURN
363      000734      101$: MSG$R 1L      ; ILLEGAL SECONDARY CSR VALUE
364      000742      111$: MSG$R 1M      ; SECONDARY CSR NOT ALLOWED
365      000750      121$: MSG$R ZO      ; SECONDARY CSR OFFLINE
366      ;
367      ; NO SECONDARY CSR (UNT$DF)
368      ;
369      ;
370      000756 032767 000000G 000000G U.XCSR: BIT    #FL.LMC,$FLAGS ; COMIOP DEVICE?
371      000764 001410      BEQ    10$      ; IF EQ, NO
372      000766 032767 000000G 000000G BIT    #FL.KMX,$FLAGS ; COMIOP MUX DEVICE?
373      000774 001404      BEQ    10$      ; IF EQ, NO
374      000776 032767 000007 000000G BIT    #7,UCNT      ; IS THIS THE FIRST, NINTH, ETC.?
375      001004 001403      BEQ    101$      ; IF EQ, YES
376      001006 005267 000000G 10$: INC    UCNT      ; INCREMENT UNIT COUNT
377      001012      RETURN
378      001014      101$: MSG$R 1N      ; SECONDARY CSR MISSING
379      ;
380      ; UNIT COST (UNT$DF)
381      ;
382      ;
383      001022 026727 000000G 000031 U.CST: CMP    .PNUMB,#MAXCST ; IS THE COST IN RANGE ?
384      001030 101026      BHI    101$      ; IF HI, NO .. ERROR
385      001032 016700 000000G      MOV    $SLTA,R0      ; GET THE SLT ADDRESS
386      001036 016701 000000G      MOV    UCNT,R1      ; GET UNIT COUNT
387      001042 005301      DEC    R1      ; UCNT HAS ALREADY BEEN INCREMENTED
388      001044 126001 000013      CMPB   L.UNIT(R0),R1      ; CORRECT UNT$DF ?
389      001050 001015      BNE    20$      ; IF NO - BRANCH
390      ;
391      001052 010001      MOV    R0,R1      ; COPY SLT ADDRESS
392      001054 032761 000400 000000      BIT    #LF.BRO,L.FLG(R1) ; BROADCAST LINE ?

```

.SBTTL CONSTANT DEFINITIONS

72  
73  
74  
75  
76  
77  
78  
79  
80

;;  
;; CONSTANT DEFINITIONS  
;;

000007  
000001  
001130  
001130

X2WNMX = 7  
X2WNMN = 1  
X2RTMX = 600.  
X2HLMX = 600.

; MAXIMUM WINDOW SIZE FOR X2P\$DF  
; MINIMUM WINDOW SIZE FOR X2P\$DF  
; MAXIMUM VALUE FOR RETRANSMIT VALUE  
; MAXIMUM VALUE FOR HOLDBACK TIMER

.TITLE ALCMEM - ALLOCATE A BLOCK OF MEMORY  
.IDENT /V05.00/

COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NETWORK BUFFER POOL ALLOCATION/DE-ALLOCATION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 27-FEB-78  
VERSION 2.0 RELEASE
- 2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

54
55
56
57      ;+ $ALL16 - ALLOCATE SINGLEWORD ADDRESSABLE STORAGE ( SCOM )
58
59      : INPUTS:
60          R1 -    BYTE COUNT NEEDED
61
62      : OUTPUTS:
63          C-BIT - SUCCESS/FAILURE
64          R0 -    BUFFER ADDRESS
65          ALL OTHER REGISTERS PRESERVED
66
67      $ALL16::
68          .IF DF R$$11M
69          SWSTK$ 20$          ;* ENTER KERNEL MODE
70
71          .IF NDF R$$MPL
72
73      000004 016700 000000G      MOV    CEAVL,R0          ;* GET CEX POOL ADDRRES
74      000010 162700 000002      SUB    #2,R0          ;* ...
75      000014 022700 177776      CMP    #-2,R0          ;* ANY POOL LEFT ?
76      000020 001403              BEQ    5$              ;* IF NO BRANCH
77      000022              CALL    @ALOC1          ;* TRY TO ALLOCATE FROM CEX POOL
78      000026 103011              BCC    10$          ;* IF CC, SUCCESS
79
80          .ENDC      ; NDF R$$MPL
81
82      000030      5$: CALL    @ALOCB          ;* TRY TO ALLOCATE FROM RSX POOL
83      000034 103006              BCC    10$          ;* IF CC, SUCCESS
84      000036 017700 000000G      MOV    @HEADR,R0      ;* GET USER MODE SP
85      000042 011000              MOV    (R0),R0          ;* ...
86      000044 005260 000006      INC    6(R0)          ;* SET USER MODE C-BIT
87      000050 000402              BR     20$          ;* LEAVE R0 UNCHANGED
88      000052 010066 000002      10$: MOV    R0,2(SP)      ;* RETURN BUFFER ADDRESS IN R0
89      000056              20$: RETURN          ;* BACK TO USER MODE AND THEN TO CALLER
90
91      .ENDC

```

```

121 .SBTTL TPARS STATE TABLES
122
123 000052 ISTAT$ CNTST,CNTKW
124
125 : CONTROLLER DEFINITION (CNT$DF)
126 :
127 STATES$ CNTDF
128 TRANS$ %CNT$DF%,1,SYNERR
129 STATES$ ; CONTROLLER NUMBER
130 TRANS$ $NUMBR,,C.CTL
131 STATES$
132 TRANS$ <','>
133 STATES$ ; VECTOR
134 TRANS$ $NUMBR,,C.VECT
135 STATES$
136 TRANS$ <','>
137 STATES$ ; CSR
138 TRANS$ $NUMBR,,C.CSR
139 STATES$
140 TRANS$ <','>
141
142 .IF NDF R$$MPL
143
144 STATES$ ; INTERRUPT PRIORITY
145 TRANS$ $NUMBR,$EXIT,C.PRI
146
147 .IFF
148
149 STATES$ ; INTERRUPT PRIORITY
150 TRANS$ $NUMBR,,C.PRI
151 STATES$
152 TRANS$ <','>
153 STATES$ ; UNIBUS RUN MASK
154 TRANS$ $NUMBR,RCDV,C.URM,1,URMFL
155 TRANS$ $LAMDA ; ALLOW FOR NON-MULTIPROCESSOR SYSTEMS
156 STATES$ RCDV
157 TRANS$ <','>,,MP.URM
158 STATES$ ; RECONFIGURATION DEVICE
159 TRANS$ $ALPHA,,C.RCD1
160 STATES$
161 TRANS$ $ALPHA,$EXIT,C.RCD2
162 .ENDC
163
164 000052 STATES$
    
```

147 000052  
148 000052  
149  
150 000052

TRANS <'>,\$EXIT  
TRANS \$EOS,\$EXIT  
STATES

CFG  
TPA

CFG



```

169          .SBTTL  TPARS STATE TABLES
170          ;
171          ; TPARS STATE TABLES
172          ;
173          ISTAT$  DTEST,DTEKW
174          ;
175          ; DTE$DF
176          ;
177          STATES$ DTEDF
178          TRANS$  %DTE$DF%,1,SYNERR
179          ;
180          STATES$
181          TRANS$  $DIGIT,,DTEAST          ; START OF DTE ADDRESS
182          ;
183          STATES$ DTEAD
184          TRANS$  $DIGIT,DTEAD,DTEADIG    ; DTE ADDRESS
185          TRANS$  <','>,,DTEAEN          ; END OF DTE ADDRESS
186          ;
187          STATES$
188          TRANS$  !LINID                  ; LINE-ID
189          ;
190          STATES$
191          TRANS$  <','>
192          ;
193          STATES$
194          TRANS$  $NUMBR,,HSHTBL          ; HASH TABLE SIZE
195          ;
196          STATES$
197          TRANS$  <','>
198          ;
199          STATES$
200          TRANS$  $NUMBR,,CNTTIM          ; COUNTER TIMER VALUE
201          ;
202          STATES$
203          TRANS$  !END,$EXIT,DTEEND
204          TRANS$  <','>
205          ;
206          STATES$
207          TRANS$  'ON',DTENW,,DE.ON,DTEFLG ; STATE
208          TRANS$  'OFF',DTENW,,DE.OFF,DTEFLG
209          ;
210          STATES$ DTENW
211          TRANS$  !END,$EXIT,DTEEND
212          TRANS$  <','>
213          ;
214          STATES$
215          TRANS$  $STRNG,,DTENET          ; NETWORK NAME PARAMETER
216          ;
217          STATES$ END
218          TRANS$  $EOS,$EXIT
219          TRANS$  <','>,$EXIT

```

CFGDTE CREATED BY MACRO ON 29-JUN-85 AT 00:03 PAGE 1 K 6  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                         |
|---------|------------|------------------------------------|
| BIASX   | 000500 R   | #17-409                            |
| CERR    | = ***** GX | 8-129 8-130 8-131 8-132 8-133      |
| CFGBF   | = ***** GX | 9-156                              |
| CFGSZ   | = ***** GX | 9-155                              |
| CFLIN   | = ***** GX | 8-129 8-130 8-131 8-132 8-133      |
| CHKDTE  | 000360 R   | 15-344 #16-368                     |
| CNTTIM  | 000264 R   | #13-312                            |
| COUNT   | 000035 R   | *7-117 *12-254 *19-470             |
| CVTBUF  | 000036 R   | *7-118 *19-462 19-466 20-490       |
| DEVCTL  | 000546 R   | #18-430                            |
| DEVNAM  | 000520 R   | #18-421                            |
| DEVUNT  | 000564 R   | #18-437                            |
| DTEADD  | 000000 R   | *7-112 12-250 12-255 12-276        |
| DTEAEN  | 000140 R   | *12-274                            |
| DTEALN  | = 000017   | *7-111 7-112 12-249 12-275 16-381  |
| DTEAST  | 000052 R   | #12-248                            |
| DTEDES  | = ***** GX | *16-376                            |
| DTEDF   | 000000 R   | 9-152                              |
| DTEDIG  | 000114 R   | #12-262                            |
| DTEEND  | 000342 R   | 14-333 #15-344                     |
| DTEFLG  | = ***** GX | *12-248                            |
| DTEKW   | 000000 RG  | 9-154 #10-173                      |
| DTEND   | = 000016 R | *7-113 12-262                      |
| DTENET  | 000302 R   | #14-325                            |
| DTEPCK  | 000017 R   | *7-114 12-277 16-384               |
| DTEST   | 000000 RG  | #10-173                            |
| DTNMMX  | = 000006   | *7-115 7-116 12-283 14-326         |
| DTNTNM  | 000027 R   | *7-116 12-282 14-329               |
| FMT8B   | = ***** GX | 8-129 8-130 8-131 8-132 8-133      |
| FM.8B   | = 000000   | #8-129 #8-130 #8-131 #8-132 #8-133 |
| HSHMN   | = 000040   | #6-84 13-295                       |
| HSHMX   | = 001000   | #6-85 13-299                       |
| HSHSZ   | = ***** GX | *13-302                            |
| HSHTBL  | 000214 R   | #13-292                            |
| HSLDTE  | 000002     | 16-369                             |
| H\$NETW | 000024     | 12-281                             |
| ISSAS   | = *****    | 20-500                             |
| KSAR5   | = ***** GX | 16-372 16-391                      |
| LF.ACT  | = 100000   | #5-66                              |
| LF.BRO  | = 000400   | #5-66                              |
| LF.BWT  | = 000007   | #5-66                              |
| LF.ENA  | = 002000   | #5-66                              |
| LF.LPB  | = 001000   | #5-66                              |
| LF.MDC  | = 000100   | #5-66                              |
| LF.MFL  | = 004000   | #5-66                              |
| LF.MTP  | = 000020   | #5-66                              |
| LF.PAC  | = 000200   | #5-66                              |
| LF.RDY  | = 040000   | #5-66                              |
| LF.REA  | = 010000   | #5-66                              |
| LF.SER  | = 000040   | #5-66                              |
| LF.TIM  | = 000010   | #5-66                              |
| LF.UNL  | = 020000   | #5-66                              |

CFGFEA      CREATED BY    MACRO    ON 29-JUN-85 AT 00:03      PAGE 2      K 7

MACRO CROSS REFERENCE      CREF    04.00

MACRO NAME      REFERENCES

|         |        |        |        |        |        |        |        |        |        |        |  |  |  |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--|--|--|
| CALL    | 6-92   |        |        |        |        |        |        |        |        |        |  |  |  |
| DBGTP\$ | #7-106 |        |        |        |        |        |        |        |        |        |  |  |  |
| EMSG\$R | #5-60  | 6-100  |        |        |        |        |        |        |        |        |  |  |  |
| ISTAT\$ | #5-60  | 7-106  |        |        |        |        |        |        |        |        |  |  |  |
| MTRANS  | #7-106 |        |        |        |        |        |        |        |        |        |  |  |  |
| REJ\$   | #5-68  | 8-146  | 8-153  |        |        |        |        |        |        |        |  |  |  |
| RETURN  | 6-96   | 8-147  | 8-154  | 8-159  | 8-164  | 8-169  |        |        |        |        |  |  |  |
| STATE\$ | #5-60  | 7-111  | #7-114 | #7-117 | #7-120 | #7-123 | #7-126 | #7-129 | #7-132 | #7-135 |  |  |  |
|         | #7-138 |        |        |        |        |        |        |        |        |        |  |  |  |
| TRAN\$  | #5-60  | #7-112 | #7-115 | #7-118 | #7-121 | #7-124 | #7-127 | #7-130 | #7-133 | #7-136 |  |  |  |

K 8

```

403
404
405      ; END OF PVC$DF MACRO
406
407 000400 016701 000026'  EVEND:  MOV    DLMSTA,R1      ; GET STATION NUMBER
408 000404 006301          ASL      R1              ; MAKE IT A WORD INDEX
409 000406 016167 000000G 000000'  MOV    $$HTIM(R1),HTIMR ; GET HELLO TIMER
410 000414 016167 000000G 000002'  MOV    $$LTIM(R1),LTIMR ; AND LISTEN TIMER
411 000422 032767 000001 000032'  BIT     #PF.DLM,FLAG    ; IS THIS A DLM PVC?
412 000430 001005          BNE     20$           ; BR IF YES
413 000432          10$:  CALL    CHKPVC          ; CHECK FOR VALID PVC ID
414 000436 103524          BCS     101$          ; BR IF ERROR
415 000440 012701 000016          MOV    #C$LEN,R1 ; GET LENGTH OF BLOCK TO ALLOCATE
416
417 000444          20$:  SWSTK$ 120$           ; ENTER SYSTEM STATE
418 000450          SAVMAP          ; SAVE PREVIOUS MAPPING
419 000454 032767 000001 000032'  BIT     #PF.DLM,FLAG    ; IS THIS A DLM PVC?
420 000462 001041          BNE     80$           ; BR IF YES
421 000464          CALL    $XALOC          ; ALLOCATE A BLOCK FROM NETWORK POOL
422 000470 103006          BCC     60$           ; BR IF SUCCESSFUL ALLOCATION
423 000472          RETC     R0              ; ELSE SET USER C-BIT
424 000504 000474          BR      110$          ; AND EXIT
425 000506 010005          60$:  MOV    R0,R5      ; SAVE UNMAPPED ADDRESS OF ALLOCATED BLOCK
426 000510 010046          MOV    R0,-(SP)      ; SET UP UNMAPPED ADDRESS FOR CONVERSION
427 000512          CALL    $CEACK          ; CONVERT TO APR5 MAPPED ADDRESS
428 000516 012600          MOV    (SP)+,R0      ; RETRIEVE MAPPED ADDRESS
429 000520 010004          MOV    R0,R4          ; SAVE MAPPED ADDRESS
430 000522 062700 000002          ADD    #C$NAM,R0 ; POINT TO CIRCUIT ID FIELD
431 000526 012702 000006          MOV    #P$NAMX,R2 ; GET LENGTH OF CIRCUIT ID
432 000532 012701 000036'          MOV    #PVCNAM,R1 ; POINT TO SPECIFIED CIRCUIT ID
433 000536 112120          70$:  MOVB   (R1)+,(R0)+ ; STORE CIRCUIT ID
434 000540 005302          DEC     R2           ; MORE TO STORE?
435 000542 003375          BGT     70$           ; BR IF YES
436 000544 116764          MOVB   PVCFLG,C$FLG(R4) ; STORE FLAGS BYTE
437 000552 016764          MOV    PVCLCN,C$LCN(R4) ; STORE LOGICAL CHANNEL NUMBER
438 000560 016764          MOV    DTEDES,C$DTE(R4) ; STORE DTE DESCRIPTOR ADDRESS
439 000566          80$:  CALL    FNDPRT          ; FIND FREE PORT ENTRY
440 000572 103006          BCC     90$           ; BR IF FOUND ONE
441 000574          RETC     R0              ; ELSE SET USER C-BIT
442 000606 000421          BR      100$          ; AND EXIT
443 000610 032767 000001 000032'  90$:  BIT     #PF.DLM,FLAG    ; IS THIS A DATA LINK MAPPING PVC?
444 000616 001403          BEQ     95$           ; BR IF NO
445 000620          CALL    DLMTLC          ; ELSE STORE PORT NUMBER IN COUNTER BLOCK
446 000624 000402          BR      97$           ;
447 000626 110164 000014          95$:  MOVB   R1,C$PORT(R4) ; STORE PORT NUMBER
448 000632          97$:  CALL    ALLXCB          ; ALLOCATE X25 CIRCUIT BLOCK
449 000636 103005          BCC     100$          ; BR IF SUCCESS
450 000640          RETC     R0              ; ELSE SET USER C-BIT
451 000652 032767 000001 000032'  100$:  BIT     #PF.DLM,FLAG    ; IS THIS A DLM PVC?
452 000660 001006          BNE     110$          ; BR IF YES - NO PVC NAME BLOCK CREATED
453 000662 017700 000000G          MOV    @PSIPT,R0 ; POINT TO PSI HOME BLOCK
454 000666 062700 000006          ADD    #H$PVC,R0 ; POINT TO PVC BLOCK LISTHEAD
455 000672          CALL    LNKEND          ; LINK CURRENT BLOCK (IN R5) TO LIST
456 000676          110$:  RESMAP          ; RESTORE PREVIOUS MAPPING
457 000702          RETURN
458
459 000704 103404          120$:  BCS     102$          ; BR IF ERROR

```

L 8

|                 |                    |                  |                 |                   |
|-----------------|--------------------|------------------|-----------------|-------------------|
| PF\$FM2= 000200 | PVSTA 002000R      | SF.ENA= 000100   | V\$LEN 000022   | X\$TIMR 000000    |
| PF\$OFF= 000000 | PVWSZ 000266R      | SF.LPB= 000004   | V\$RCV 000002   | X\$TXI 000102     |
| PF\$RM1= 000020 | PX\$BLK= 000040    | SF.MFL= 000040   | V\$XMT 000012   | X\$TXQ 000066     |
| PF\$RM2= 000040 | PX\$DLM= 000200    | SF.PAC= 000020   | V\$SCTR= 001000 | X\$TYP 000015     |
| PF\$RVR= 000010 | PX\$SVC= 000100    | SF.REA= 000010   | W\$DSZ 000066R  | X\$USR 000022     |
| PF\$STA= 000004 | P\$CHN 000004      | SF.SER= 000001   | X\$SNCO= 000020 | X\$WAQ 000062     |
| PF\$SVC= 040000 | P\$CNT 000005      | SF.SVC= 000002   | X\$SRR= 000200  | X\$WSZ 000020     |
| PF\$UP= 000004  | P\$CTR 000034      | SF.UNL= 000040   | X\$SUCL= 000040 | X\$SDBT= 000000   |
| PF.DLM= 000001  | P\$DRCT 000015     | SLTMA= ***** GX  | X\$SURE= 000100 | ZTIME= ***** GX   |
| PORTNO 000004R  | P\$DRTR 000036     | SPACE= 000040    | X\$SNCO= 000020 | Z.DAT= ***** GX   |
| PR\$ADD 000006R | P\$FLG 000012      | SP\$P4 000050RG  | X\$SOFF= 000200 | Z.MAP= ***** GX   |
| PR\$BED= 000200 | P\$FWD 000030      | SYNERR= ***** GX | X\$SORI= 000002 | Z.NAM= ***** GX   |
| PR\$BEU= 000100 | P\$ICCB 000046     | S\$SWRG= 000000  | X\$SOTI= 000001 | \$ALPHA= 000022   |
| PR\$BRD= 000040 | P\$IPL 000014      | S\$SYSZ= 007600  | X\$STIL= 000010 | \$ANY= 000020     |
| PR\$BRU= 000020 | P\$LCD 000002      | S.COST 000001    | X\$STIR= 000004 | \$BLANK= 000006   |
| PR\$DWN= 000002 | P\$LEN 000052      | S.FLG 000000     | X\$TCHN= 000100 | \$CAT5= ***** GX  |
| PR\$LCC= 000010 | P\$LST 000000      | S.LEN 000004     | X\$TDGR= 000002 | \$CEACX= ***** GX |
| PR\$MOP= 000004 | P\$NRNI 000040     | S.NMST 000002    | X\$TFAS= 000200 | \$DIGIT= 000024   |
| PR\$UP= 000001  | P\$OCCB 000050     | S.OWNR 000003    | X\$TINC= 000004 | \$DNUMB= 000014   |
| PSIPT= ***** GX | P\$PFQ 000006      | T\$FLAG 000044   | X\$TOUT= 000010 | \$EOS= 000012     |
| PSZ 000044R     | P\$PKSZ 000044     | T\$LIIF 000013   | X\$TPVC= 000001 | \$ERRNO 000052R   |
| PS\$CHR= 000016 | P\$RMX1 000016     | T\$LIIF 000013   | X\$ABQ 000076   | \$ERRN1 000112R   |
| PS\$FAI= 000014 | P\$RMX2 000020     | T\$LIFO 000013   | X\$ALF 000055   | \$ERRN2 000146R   |
| PS\$NTI= 000006 | P\$RPRI 000042     | T\$LIFFS 000013  | X\$AUC 000056   | \$ERRN3 000200R   |
| PS\$OFF= 000000 | P\$RTIM 000003     | T\$LIN 000000    | X\$CLEN= 000025 | \$ERRR0 000300R   |
| PS\$STR= 000002 | P\$STA1 000022     | T\$LIPS 000006   | X\$CTIM 000060  | \$ERRR5 000334R   |
| PS\$UP= 000012  | P\$STA2 000023     | T\$LLD 000012    | X\$DIAG 000003  | \$ERRR9 000370R   |
| PS\$VER= 000010 | P\$TIM 000010      | T\$LLDC 000045   | X\$DTE 000024   | \$ERRP8 000244R   |
| PS\$WT= 000004  | P\$TSC 000026      | T\$LLDL 000012   | X\$FLG 000014   | \$ERR1T= ***** GX |
| PT\$BRO= 000200 | P\$TSIZ 000024     | T\$LLDO 000012   | X\$GLEN 000106  | \$EXIT= 000000    |
| PT\$DRI= 000100 | P\$TYP 000001      | T\$LLDS 000012   | X\$LCN 000026   | \$FAIL= 177777    |
| PT\$END= 000004 | P\$P45= 000000     | T\$LLN 000046    | X\$LEN 000106   | \$HEADR= ***** GX |
| PT\$LV1= 000002 | P\$WRD= 000000     | T\$LOPR 000002   | X\$MOWN 000023  | \$LAMDA= 000000   |
| PT\$LV2= 000001 | Q\$SOP7= 000010    | T\$LTCL 000024   | X\$NLRE 000052  | \$NUMBER= 000002  |
| PT\$PH3= 000010 | RF.CTL= 000003     | T\$LTJM 000026   | X\$NNRE 000054  | \$QVVC 000000RG   |
| PT\$XAR= 000020 | RF.LD1= 000040     | T\$LTJR 000014   | X\$NPL 000006   | \$RAD50= 000016   |
| PVCTM 000012R   | RF.LD2= 000100     | T\$LTPS 000020   | X\$NPR 000007   | \$SLTA= ***** GX  |
| PVCFD 000000R   | RF.TIM= 177400     | T\$NAPL 000004   | X\$NRBY 000032  | \$STRNG= 000004   |
| PVCFLG 000035R  | CJ2 RF.TMO= 000400 | T\$NFE 000000    | X\$NRPK 000042  | \$SUBXP= 000010   |
| PVCLCN 000010R  | RF.WFC= 000200     | T\$NLEN 000010   | X\$NNRE 000053  | \$TRIB= ***** GX  |
| PVCNAM 000036R  | RF.WTD= 000020     | T\$NNUL 000002   | X\$NTBY 000036  | \$XALOC= ***** GX |
| PVCOWN 000020R  | RF.WTM= 000030     | T\$NOPL 000006   | X\$NTPK 000046  | \$X3DBS= ***** GX |
| PVCPRT 000034R  | RF.WTS= 000010     | T\$NRNJ 000042   | X\$PKSZ 000016  | \$X3DWS= ***** GX |
| PVCPSZ 000014R  | RTSPC= ***** GX    | T\$NRPL 000005   | X\$PR 000011    | \$SHTIM= ***** GX |
| PVCTST 000000RG | RSR0= 000002       | T\$NRUL 000007   | X\$PRT 000021   | \$SLTIM= ***** GX |
| PVCT 000220R    | RSR1= 000004       | T\$NVR 000001    | X\$PS 000010    | \$SFLG= 177777    |
| PVCTL 001726R   | RSR2= 000006       | T\$RPRJ 000040   | X\$RPR 000013   | \$SKEY= 000002    |
| PVCWSZ 000016R  | RSR3= 000010       | T\$SVC 000034    | X\$RPS 000012   | \$S\$R= 000010    |
| PVDLM 001630R   | RSR4= 000012       | T\$T5 000030     | X\$RTRY 000002  | \$SSTA= 000000    |
| PVEND 000400P   | RSR5= 000014       | T\$T6 000032     | X\$RXI 000104   | \$S\$TMP= 000013R |
| PVFG 000346R    | R\$S\$DER= 000000  | T\$TKMG= 000000  | X\$RXQ 000072   | .PNUMB= ***** GX  |
| PVLN 000162R    | R\$SK11= 000001    | T\$SMIN= 000000  | X\$SS 000004    | .PNUMH= ***** GX  |
| PVNAM 000056R   | R\$SND= 000000     | V\$RCV= 100000   | X\$ST 000005    | .PSTCN= ***** GX  |
| PVNAMX= 000006  | R\$S11M= 000000    | V\$XMT= 040000   | X\$STCLZ 000030 | .PSTPT= ***** GX  |
| PVOWN 000310R   | SF.ACT= 000200     | V\$FLG 000000    | X\$TIMC 000001  | .TPARS= ***** GX  |
| PVPSZ 000244R   |                    |                  |                 |                   |

```

$QSLT - LOOK FOR SLT$DF MACRO

      .SBTTL $QSLT - LOOK FOR SLT$DF MACRO

      ;+
      ; $QSLT - LOOK FOR SLT$DF MACRO
      ; INPUTS:
      ;     NONE
      ; OUTPUTS:
      ;     C-BIT=SUCCESS/FAILURE
      ;     R3,R4,R5=DESTROYED
      ; -

84 000000 012705 000000' $QSLT:: MOV #SLTDF,R5 ; STATE TABLE ADDRESS
85 000004 005001          CLR R1 ; FULL KEYWORD MATCH LENGTH
86 000006 012702 000000' MOV #SLTKW,R2 ; KEYWORD TABLE ADDRESS
87 000012 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
88 000016 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
89 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
90 000026 005067 000000G CLR .NBRA ; CLEAR BROADCAST ROUTER ADJACENCIES
91 000032 005067 000000G CLR .RPRI ; CLEAR ROUTER PRIORITY
92 000036          CALL TPARS ; GO DO THE PARSE
93 000042 103003          BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
94 000044 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
95 000050 001401          BEQ 101$ ; IF EQ, YES
96 000052          20$: RETURN

      ;
      ; ERROR CONDITION
      ;
101 000054          101$: MSG$R 1T ; SYNTAX ERROR

```

65  
66  
67  
68  
69  
70

.SBTTL CONSTANT DEFINITIONS

...  
CONSTANTS  
...

000031

MAXCST = 25.

; MAXIMUM LINE COST ALLOWED

CFG SVC - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 00:06  
 Table of contents

|     |     |  |
|-----|-----|--|
| 5-  | 54  | MACRO DEFINITIONS                            |
| 6-  | 109 | LOCAL DATA                                   |
| 7-  | 135 | ERROR MESSAGES                               |
| 8-  | 152 | \$QSVC - LOOK FOR SVC\$DF MACRO              |
| 9-  | 181 | SVC\$DF STATE TABLE                          |
| 10- | 255 | SVC\$DF ACTION ROUTINES                      |
| 14- | 495 | FNDOWN - FIND SVC OWNER'S PDV INDEX          |
| 15- | 519 | SETTLC - SET UP TRANSPORT LINE COUNTERS      |
| 16- | 585 | PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT |
| 17- | 622 | STRNXT - STORE NEXT DIGIT                    |



```

FNDOWN - FIND SVC OWNER'S PDV INDEX

      .SBTTL  FNDOWN - FIND SVC OWNER'S PDV INDEX
      :+
      : FNDOWN - FIND PDV INDEX OF SVC OWNER
      : INPUTS:
      :   R2 - RAD50 PROCESS NAME OF SVC OWNER
      : OUTPUTS:
      :   CARRY CLEAR:
      :   R2 - PDV INDEX OF PROCESS
      :   CARRY SET:
      :   PDV INDEX OF OWNER NOT FOUND
      :-
      FNDOWN: SWSTK$ 20$      :: ENTER SYSTEM STATE
      MOV      R1,R2        :: GET PROCESS NAME TO LOOK FOR
      CALL     @PDVID       :: GET PDV INDEX
      BCC      10$          :: BR IF FOUND IT
      RETC     R0           :: ELSE SET USER C-BIT
      10$:     MOVB      R2,OWNSVC  :: SAVE OWNER PDV INDEX
      20$:     RETURN
  
```

\*\*FILE\*\*ID\*\*CFGUNT

K 14

```
CCCCCCCC FFFFFFFF GGGGGGGG UU UU NN NN TTTTTTTTTT
CCCCCCCC FFFFFFFF GGGGGGGG UU UU NN NN TTTTTTTTTT
CC FF GG UU UU NN NN TT
CC FF GG UU UU NN NN TT
CC FF GG UU UU NNNN NN TT
CC FF GG UU UU NNNN NN TT
CC FFFFFFFF GG UU UU NN NN TT
CC FFFFFFFF GG UU UU NN NN TT
CC FF GG GGGGGG UU UU NN NNNN TT
CC FF GG GGGGGG UU UU NN NNNN TT
CC FF GG GG UU UU NN NN TT
CC FF GG GGGGGG UU UU NN NN TT
CCCCCCCC FF GGGGGG UUUUUUUUUU NN NN TT
CCCCCCCC FF GGGGGG UUUUUUUUUU NN NN TT
```

....  
....  
....  
....

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
SS TT
SS TT
SS TT
SS TT
SS TT
SS SSSSSS TT
SS SSSSSS TT
SS SS TT
SS SS TT
SS SS TT
SS SS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

```

393 001062 001405          BEQ      10$          ; IF NO - BRANCH
394 001064 062701 000022    ADD      #L.MPF,R1      ; POINT TO STATION EXTENSION
395 001070 116761 000000G 000001  MOVB    .PNUMB,S.COST(R1) ; STORE COST
396                                     ;
397 001076 116760 000000G 000015 10$:  MOVB    .PNUMB,L.COST(R0) ; STORE UNIT COST
398 001104                                     20$:  RETURN
399                                     ;
400 001106                                     101$:  MSG$R YP          ; ILLEGAL LINE COST
401                                     ;
402                                     ; UNIT DEAD POLLING RATIO
403                                     ;
404                                     ;
405 001114                                     U.DPR:
406 001114 016767 000000G 000000G  MOV     .PNUMB,$$DPR    ; SAVE DEAD POLLING RATIO
407 001122                                     RETURN
  
```

CFGX2P - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 00:08 Page 7

LOCAL DATA

.SBTTL LOCAL DATA

82  
83  
84  
85  
86  
87  
88  
89  
90

;  
; LOCAL DATA

X2DEV: .BLKW 1  
X2CTL: .BLKB 1  
X2UNT: .BLKB 1

; DEVICE NAME  
; CONTROLLER NUMBER  
; UNIT NUMBER

CFGX2P - CONFIG FILE SCAN ACTIO MACRO V05.03b Saturday 29-Jun-85 00:08 Page 8

THESE ROUTINES WERE ADAPTED FROM THE RSX11D/IAS MEMORY ALLOCATION ROUTINES .FMEM & .FRSEG ( EM09.MAC ) FOR USE BY THE DISTRIBUTED SYSTEMS GROUP ON THEIR RSX FAMILY PRODUCTS. THE COMMENTS IN THESE MODULES ARE VERY HARD TO UNDERSTAND. THE VIRTUAL MEMORY ALLOCATION ROUTINE COMMENTS HAVE BEEN REWRITTEN AND IT IS RECOMMENDED THAT IF THE FLOW OF THIS CODE MUST BE FOLLOWED, THAT THOSE COMMENTS BE USED AS A GUIDE ( VAL22.MAC ).

THE GLOBAL SYMBOL \$QBIAS IS THE LISTHEAD FOR A LINKED LIST OF FREE SEGMENTS OF MEMORY. THE CHAIN POINTERS ARE KEPT IN THE FIRST WORD OF EACH SEGMENT AND ARE THE REAL MEMORY ADDRESS MODULO 64 ( 1/64 REAL MEMORY ADDRESS). THIS IS THE VALUE THAT SHOULD BE PUT INTO THE SCRATCH APR TO MAP TO THE NEXT SEGMENT. THE SECOND WORD OF EACH SEGMENT IS THE SIZE IN 32 WORD BLOCKS OF THE CURRENT SEGMENT. IT IS IMPORTANT TO UNDERSTAND THAT ALL SEGMENTS MUST BE ALLIGNED ON 32 WORD BOUNDARIES, BUT THAT THE LISTHEAD \$QBIAS IS NOT SO CONSTRAINED. THIS CAUSES SOME EXCEPTION TESTING IN THESE ROUTINES.

INITIALLY \$QBIAS SHOULD CONTAIN THE BIAS ( 1/64 PHYSICAL MEMORY ADDRESS ) FOR THE BASE OF THE MEMORY TO BE AS A BUFFER POOL. THIS POOL CAN BE ALLOCATED IN THE SYSTEM BY EITHER USING A PARTITION OR TASK. ADDITIONALLY THE FIRST WORD OF THE PARTITION/TASK SHOULD BE ZERO ( LAST SEGMENT ) AND THE SECOND WORD THE SIZE OF THE PARTITION/TASK IN 32-WORD BLOCKS.

INITIALLY:

|             |                             |
|-------------|-----------------------------|
| \$QBIAS --> | 0                           |
|             | SIZE                        |
|             | REST<br>OF<br>FREE<br>SPACE |

AFTER THESE TWO ROUTINES HAVE BEEN RUNNING FOR AWHILE, THE THE PARTITION/TASK WILL PROBABLY CONTAIN A LINKED LIST OF FREE SEGMENTS OF MEMORY. THE LISTHEAD \$QBIAS IS MAINTAINED BY THESE ROUTINES SO THAT IT IS ALWAYS A POINTER TO THE FIRST FREE SEGMENT OF MEMORY. ONE SURE-FIRE WAY TO CRASH THE SYSTEM IS TO MODIFY \$QBIAS SINCE THIS ROUTINE CANNOT TELL WHETHER OR NOT IT IS PROCESSING GARBAGE AND CAN ALLOCATE MEMORY FROM ANYWHERE IN THE REAL ADDRESS SPACE. THESE ROUTINES ALWAYS AGGLOMERATE SEGMENTS WHEN THEY ARE CONTIGUOUS. THESE ROUTINES ARE NOT REENTRANT. NOTE THAT IT IS THE CALLERS' RESPONSIBILITY TO SET UP THE ADDRESS BIAS AND SIZE OF SEGMENTS BEING RETURNED TO THE POOL CORRECTLY. IF EITHER VALUE IS INCORRECT, IT

```

93      ;+
94      ; $DEA16 - DE-ALLOCATE SINGLEWORD ADDRESSABLE STORAGE ( SCOM )
95      ;
96      ; INPUTS:
97      ;   R0 -   BUFFER ADDRESS
98      ;   R1 -   BYTE COUNT
99      ;
100     ; OUTPUTS:
101     ;   C-BIT WILL BE SET IF BUFFER ADDRESS IS ILLEGAL.
102     ;
103     ; -
104     ; .ENABL  LSB
105     $DEA16::
106     ; .IF DF  R$$11M
107     ; SWSTK$ 20$          ; * ENTER KERNEL MODE
108
109     $DECEX::
110     ; .IF NDF R$$MPL
111
112     000064 016703 000000G      MOV     CEAVL,R3          ; * ASSUME CEX POOL
113     000070 162703 000002      SUB     #2,R3           ; * GET FREE POOL LISTHEAD
114     000074 027700 000000G      CMP     @EXSIZ,R0        ; * IS BUFFER FROM CEX POOL?
115     000100 101404              BLOS    10$             ; * IF LOS, YES
116
117     ; .ENDC
118
119     000102 016703 000000G      MOV     CRAVL,R3          ; * ELSE USE RSX POOL
120     000106 162703 000002      SUB     #2,R3           ; * GET FREE POOL LISTHEAD
121     000112      10$: CALL     @DEAC1          ; * RETURN BUFFER TO APPROPRIATE POOL
122     000116      20$: RETURN          ; * BACK TO USER MODE AND THEN TO CALLER
123     ; .ENDC
124
125     ; .DSABL  LSB

```

```

166 .SBITL CNT$DF ACTION ROUTINES
167
168 ; CONTROLLER NUMBER (CNT$DF)
169
170 000052 126767 000000G 000000G C.CTL: CMPB .PNUMB,CTL ; REJECT TRANSITION IF NO MATCH
171 000060 001404 BEQ 10$ ; ...
172 000062 REJ$ ; ...
173 000072 10$: RETURN ; ...
174
175 ; INTERRUPT VECTOR (CNT$DF)
176
177 000074 032767 000000G 000000G C.VECT: BIT #MS.VCT,$MISS ; IS VECTOR MISSING?
178 000102 001421 BEQ 10$ ; IF EQ, NO
179 000104 005767 000000G TST .PNUMH ; MAKE SURE IT'S A VALID VECTOR
180 000110 001017 BNE 101$ ; ...
181 000112 032767 000003 000000G BIT #3,.PNUMB ; ...
182 000120 001013 BNE 101$ ; ...
183 000122 026767 000000G 000000G CMP .PNUMB,$MXVCT ; AND THAT IT'S IN OUR SYSTEM
184 000130 103012 BHIS 111$ ; ...
185 000132 016767 000000G 000000G MOV .PNUMB,$$VECT ; STORE VECTOR VALUE
186 000140 042767 000000G 000000G BIC #MS.VCT,$MISS ; IT'S NO LONGER MISSING
187 000146 10$: RETURN
188 000150 101$: MSG$R 1C ; ILLEGAL VECTOR VALUE
189 000156 111$: MSG$R 1D ; VECTOR NOT IN SYSTEM
190
191 ; CSR (CNT$DF)
192
193
194 000164 032767 000000G 000000G C.CSR: BIT #MS.CSR,$MISS ; IS CSR MISSING?
195 000172 001413 BEQ 10$ ; .. NO
196 000174 CALL GETCSR ; GET CSR ADDRESS
197 000200 103411 BCS 101$ ; .. ILLEGAL VALUE
198
199 .IF DF R$$MPL
200
201 ISTB @NCPU ; IS THIS A MULTI-PROCESSOR SYSTEM ?
202 BNE 5$ ; IF NE, DON'T CHECK CSR - YET
203
204 .ENDC
205
206 000202 CALL $NLC SR ; TEST FOR ONLINE
207 000206 103411 BCS 111$ ; .. OFFLINE
208 000208 010067 000000G 5$: MOV RO,$$CSR ; STORE CSR VALUE
209 000210 042767 000000G 000000G BIC #MS.CSR,$MISS ; IT'S NO LONGER MISSING
210 000222 10$: RETURN
211 000224 101$: MSG$R 1E ; ILLEGAL CSR VALUE
212 000232 111$: MSG$R 22 ; DEVICE OFFLINE
213
214 ; INTERRUPT PRIORITY (CNT$DF)
215
216 000240 032767 000000G 000000G C.PRI: BIT #MS.PRI,$MISS ; IS PRIORITY MISSING?
217 000246 001424 BEQ 10$ ; IF EQ, NO
218 000250 005767 000000G TST .PNUMH ; MAKE SURE IT'S A VALID PRIORITY
219 000254 001022 BNE 101$ ; ...
220 000256 026727 000000G 000007 CMP .PNUMB,#7 ; ...
221 000264 101016 BHI 101$ ; ...
222 000266 016700 000000G MOV .PNUMB,RO ; GET PRIORITY VALUE

```

```

152 .SBTTL ACTION ROUTINES
153
154 ; PROCESS CONTROLLER TABLE COUNT (DDM$DF, DLC$DF)
155 ;
156 .IF DF M$$MGE
157 000052 D.EXT: BIT #MS.EXT,$MISS ; IS THE CONTROLLER TABLE COUNT MISSING ?
158 000052 032767 000000G 000000G BEQ 10$ ; IF EQ, NO - LEAVE
159 000060 001403 000000G 000000G MOV .PNUMB,$$EXT ; GET THE CONTROLLER TABLE COUNT
160 000062 016767 000000G 000000G 10$: RETURN
161 000070 .ENDC
162
163
164 000001 .END
  
```



```
221 .SBTTL LINE-ID SUBEXPRESSION
222
223 ; LINE-ID SUBEXPRESSION
224 ;
225
226 000052 STATES$ LINID
227 000052 TRANS$ $RAD50,,DEVNAM ; DEVICE NAME
228
229 000052 STATES$
230 000052 TRANS$ <'->
231
232 000052 STATES$
233 000052 TRANS$ $NUMBR,,DEVCTL ; CONTROLLER NUMBER
234
235 000052 STATES$
236 000052 TRANS$ <'>,,LINID1
237 000052 TRANS$ $LAMDA,$EXIT
238
239 000052 STATES$ LINID1
240 000052 TRANS$ $NUMBR,$EXIT,DEVUNT ; UNIT NUMBER
241
242 000052 STATES$
```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                           |
|---------|------------|--------------------------------------|
| LF.X2P  | = 000000   | #5-66                                |
| LINCTL  | 000044 R   | #7-122 *18-432                       |
| LINNAM  | 000042     | #7-121 *18-425                       |
| LINUNT  | 000045 R   | #7-123 *18-439                       |
| LN.CLO  | = 000000   | #5-66                                |
| LN.DUM  | = 000005   | #5-66                                |
| LN.LOA  | = 000004   | #5-66                                |
| LN.LOO  | = 000003   | #5-66                                |
| LN.OAU  | = 000003   | #5-66                                |
| LN.OFF  | = 000001   | #5-66                                |
| LN.ON   | = 000000   | #5-66                                |
| LN.OOP  | = 000004   | #5-66                                |
| LN.OPE  | = 000001   | #5-66                                |
| LN.REF  | = 000002   | #5-66                                |
| LN.SER  | = 000002   | #5-66                                |
| LN.STA  | = 000017   | #5-66                                |
| LN.SUB  | = 000360   | #5-66                                |
| LN.TRI  | = 000006   | #5-66                                |
| L\$DTEA | 000020     | 16-383                               |
| L.COST  | 000015     | #5-66                                |
| L.CTL   | 000012     | #5-66                                |
| L.CVA   | 177776     | #5-66                                |
| L.DDM   | 000002     | #5-66                                |
| L.DDS   | 000004     | #5-66                                |
| L.DLC   | 000003     | #5-66                                |
| L.DLM   | 000006     | #5-66                                |
| L.DLS   | 000010     | #5-66                                |
| L.FLG   | 000000     | #5-66                                |
| L.KRBA  | 000016     | #5-66                                |
| L.LEN   | = 000022   | #5-66                                |
| L.MPF   | 000022     | #5-66                                |
| L.NMST  | 000020     | #5-66                                |
| L.NSTA  | 000014     | #5-66                                |
| L.OWNR  | 000021     | #5-66                                |
| L.UNT   | 000013     | #5-66                                |
| NEXT    | 000040 R   | #7-120 *12-255 12-262 19-468 *19-469 |
| NUMDTE  | = 000020   | #6-86                                |
| N\$SVCT | = *****    | 16-390                               |
| PCKBCD  | 000660 R   | 12-278 #20-489                       |
| PSIPT   | = ***** GX | 12-280 16-368                        |
| RTSPC   | = ***** GX | 8-129 8-130 8-131 8-132 8-133        |
| R\$RO   | = 000002   | #6-91                                |
| R\$R1   | = 000004   | #6-92                                |
| R\$R2   | = 000006   | #6-93                                |
| R\$R3   | = 000010   | #6-94                                |
| R\$R4   | = 000012   | #6-95 *16-392                        |
| R\$R5   | = 000014   | #6-96                                |
| R\$SEIS | = *****    | 20-500                               |
| R\$S110 | = *****    | 20-500                               |
| SF.ACT  | = 000200   | #5-66                                |
| SF.ENA  | = 000100   | #5-66                                |
| SF.LPR  | = 000004   | #5-66                                |

\*\*FILE\*\*ID\*\*CFGPVC

L 7

```
CCCCCCCC FFFFFFFF GGGGGGGG P P P P P P V V V V CCCCCCCC
CCCCCCCC FFFFFFFF GGGGGGGG P P P P P P V V V V CCCCCCCC
CC FF GG PP PP VV VV CC
CC FF GG PP PP VV VV CC
CC FF GG PP PP VV VV CC
CC FFFFFFFF GG P P P P P P VV VV CC
CC FFFFFFFF GG P P P P P P VV VV CC
CC FF GG GGGGGG PP VV VV CC
CC FF GG GGGGGG PP VV VV CC
CC FF GG GG GG PP VV VV CC
CC FF GG GG GG PP VV VV CC
CCCCCCCC FF GGGGGG PP VV VV CCCCCCCC
CCCCCCCC FF GGGGGG PP VV VV CCCCCCCC
```

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

460 000706  
461  
462  
463  
464 000710  
465 000716

RETURN  
: ERRORS  
101\$: MSG\$R N0 ; INVALID PVC ID  
102\$: MSG\$R N3 ; PVC/XCB/PORT/HASH ALLOCATION FAILED

CFGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:04 Page 21-4  
Symbol table

|         |        |     |                    |
|---------|--------|-----|--------------------|
| . ABS.  | 177776 | 000 | (RW,I,GBL,ABS,OVR) |
|         | 002272 | 001 | (RW,I,LCL,REL,CON) |
| DATA    | 000430 | 002 | (RW,D,LCL,REL,CON) |
| \$STATE | 000212 | 003 | (RW,D,LCL,REL,CON) |
| \$KTAB  | 000006 | 004 | (RW,D,LCL,REL,CON) |
| \$KSTR  | 000016 | 005 | (RW,D,LCL,REL,CON) |

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 94  
Work file writes: 93  
Size of work file: 23564 Words ( 93 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:01:07.15  
SY:CFGPVC.V2,[132,134]CFGPVC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFGPVC

```

103          .SBTTL  TPARS TABLES
104          :
105          : TPARS STATE TABLES
106          :
107 000062          ISTAT$  SLTST,SLTKW
108
109          :
110          : SLT DEFINITION MACRO (SLT$DF)
111          :
112 000062          STATES$  SLTDF
113 000062          TRANS$  %SLT$DF%,...1,SYNERR
114 000062          STATES$          ; DDM NAME
115 000062          TRANS$  $RAD50,,S.NAME
116 000062          STATES$
117 000062          TRANS$  <'>
118 000062          STATES$          ; DLC NAME
119 000062          TRANS$  $RAD50
120 000062          STATES$
121 000062          TRANS$  <'>
122 000062          STATES$          ; LLC NAME
123 000062          TRANS$  $RAD50
124 000062          STATES$
125 000062          TRANS$  <'>
126 000062          STATES$          ; FLAGS
127 000062          TRANS$  !BITS
128 000062          STATES$          ; CONTROLLER NUMBER
129 000062          TRANS$  $NUMBR,,S.CTL
130 000062          STATES$
131 000062          TRANS$  <'>
132 000062          STATES$          ; UNIT NUMBER
133 000062          TRANS$  $NUMBR,,S.UNT
134 000062          STATES$
135 000062          TRANS$  !END,$EXIT
136 000062          TRANS$  <'>
137 000062          STATES$
138 000062          TRANS$  <'>,SCTIM
139 000062          TRANS$  $STRNG          ; TYPE OF LINE
140 000062          STATES$
141 000062          TRANS$  !END,$EXIT
142 000062          TRANS$  <'>
143 000062          STATES$  SCTIM
144 000062          TRANS$  $NUMBR,,SCHELO,S.CTIM ; COUNTER TIMER
145 000062          TRANS$  <'>,SCHEL
146 000062          STATES$  SCHELO
147 000062          TRANS$  !END,$EXIT
148 000062          TRANS$  <'>
149 000062          STATES$  SCHEL
150 000062          TRANS$  $NUMBR,BRADJ,S.HTIM ; HELLO LISTEN TIMER
151 000062          TRANS$  <'>
152 000062          STATES$  BRADJ
153 000062          TRANS$  <'>,BRADJ1
154 000062          TRANS$  !END,$EXIT
155 000062          STATES$  BRADJ1
156 000062          TRANS$  $NUMBR,,S.BRAD
157 000062          STATES$
158 000062          TRANS$  <'>
159 000062          STATES$
  
```

ERROR MESSAGES

.SBTTL ERROR MESSAGES

72  
73  
74  
75 000000  
76 000030  
77 000070  
78 000130  
79 000162  
80

.ENABL LC  
NTLRS\$ ,1A,8B,CERR,RTSPC,CFLIN,<Station Address>  
NTLRS\$ ,1B,8,CERR,RTSPC,CFLIN,<More Than 48. STA\$DF's>  
NTLRS\$ ,YQ,8B,CERR,RTSPC,CFLIN,<Multipoint Active Ratio>  
NTLRS\$ ,YA,8B,CERR,RTSPC,CFLIN,<hello timer value>  
NTLRS\$ ,YB,8B,CERR,RTSPC,CFLIN,<listen timer value>  
.DSABL LC

.TITLE CFG SVC - CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 04-JUN-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0



L 13

.SBTTL SETTLC - SET UP TRANSPORT LINE COUNTERS

SETTLC - SET UP TRANSPORT LINE COUNTERS

INPUTS:

R0 - APR BIAS OF SVC DESCRIPTOR BLOCK  
 R1 - VIRTUAL ADDRESS OF SVC DESCRIPTOR BLOCK

OUTPUTS:

APR BIAS AND VIRTUAL ADDRESS OF SVC DESCRIPTOR BLOCK  
 STORED IN TRANSPORT LINE COUNTER BLOCK

R0,R1,R2,R3,R4 DESTROYED

NOTE: THIS ROUTINE MUST BE CALLED FROM SYSTEM STATE

```

519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537 001202 017703 000000G SETTLC: MOV @SLTMA,R3 ;; GET SLT VECTOR TABLE ADDRESS
538 001206 022367 000016' 5$: CMP (R3)+,SLTMA ;; IS THIS THE CORRECT ENTRY?
539 001212 001375 BNE 5$ ;; BR IF NO
540 001214 005743 TST -(R3) ;; POINT TO CORRECT ENTRY
541 001216 167703 000000G SUB @SLTMA,R3 ;; GET SYSTEM LINE NUMBER*2
542 001222 067703 000000G ADD @LLCTA,R3 ;; INDEX INTO REVERSE MAPPING TABLE
543 001226 011303 MOV (R3),R3 ;; GET CHANNEL NUMBER AND PDV INDEX
544 001230 100004 BPL 10$ ;; BR IF NOT MULTIPoint
545 001232 066703 000014' ADD STASVC,R3 ;; POINT TO CORRECT STATION ENTRY
546 001236 006303 ASL R3 ;; GET ADDRESS OF STATION TABLE
547 001240 011303 MDV (R3),R3 ;; GET CHANNEL NUMBER AND PDV INDEX
548 001242 005046 10$: CLR -(SP) ;; GET INDEX INTO FORWARD MAPPING TABLE
549 001244 110316 MOVB R3,(SP) ;; ...
550 001246 006316 ASL (SP) ;; RETRIEVE FORWARD MAPPING TABLE INDEX
551 001250 012603 MOV (SP)+,R3 ;; GET XPT'S PDV INDEX
552 001252 005046 CLR -(SP) ;; POINT TO XPT'S PDV ADDRESS
553 001254 116716 000020' MOVB @PDVTA,(SP) ;; GET XPT'S PDV ADDRESS
554 001260 067716 000000G ADD @PDVTA,(SP) ;; POINT TO FORWARD MAPPING TABLE ENTRY
555 001264 013602 MOV @SP+,R2 ;; GET SLN/STA PAIR
556 001266 060203 ADD R2,R3 ;; SAVE CURRENT MAPPING
557 001270 062703 000000G ADD #2,MAP,R3 ;; GET ADDRESS OF XPT'S DDB
558 001274 011304 MDV (R3),R4 ;; MAP TO CIRCUIT COUNTERS
559 001276 SAVMAP ;; GET NUMBER OF CIRCUIT COUNTERS
560 001302 016202 000000G MOV Z,DAT(R2),R2 ;; GET ADDRESS OF FIRST COUNTER BLOCK
561 001306 MAP NSTLC+2(R2) ;; BIAS COUNTER BLOCK ADDRESS
562 001314 016203 000100 MOV NSTLC(R2),R3 ;; HAS THIS BLOCK BEEN ASSIGNED ALREADY?
563 001320 016202 000104 MDV NSTLC+4(R2),R2 ;; BR IF NO - USE IT
564 001324 BIAS R2 ;; ELSE IS THIS THE ONE WE WANT?
565 001334 005712 20$: TST (R2) ;; BR IF YES
566 001336 100406 BMI 30$ ;; POINT TO NEXT COUNTER BLOCK
567 001340 021204 CMP (R2),R4 ;; CHECK ALL COUNTER BLOCKS
568 001342 001405 BEQ 40$ ;; STORE SLN/STA PAIR
569 001344 062702 000046 ADD #1$LEN,R2 ;; CHECK DLM SVC
570 001350 001350 SDB R3,20$ ;; INDICATE DLM SVC
571 001354 010412 30$: MOV R4,(R2) ;; ASSUME SVC DESCRIPTOR IS IN DSR
572 001356 152762 000300 000044 40$: BISB #PX$SVC!PX$DLM,T$FLAG(R2) ;; IS SVC DESCRIPTOR IN DSR?
573 001364 005062 000034 CLR T$SVC(R2) ;; BR IF YES
574 001370 020127 120000 CMP R1,#120000
575 001374 103406 BLO 50$

```

M 13

CFGUNT - CONFIG FILE SCAN ACTION MACRO V05.03b Monday 15-Jul-85 18:56 <sup>L 14</sup>  
Table of contents

|     |     |                                     |
|-----|-----|-------------------------------------|
| 5-  | 53  | MACRO DEFINITIONS                   |
| 6-  | 62  | LOCAL DATA AND CONSTANT DEFINITIONS |
| 7-  | 101 | ERROR MESSAGES                      |
| 8-  | 121 | \$QUNT - LOOK FOR UNT\$DF MACRO     |
| 9-  | 151 | UNT\$DF STATE TABLE                 |
| 11- | 210 | UNT\$DF ACTION ROUTINES             |

CFGUNT - CONFIG FILE SCAN ACTION MACRO V05.03b Monday 15-Jul-85 18:56 <sup>L 15</sup> Page 14  
UNTSDF ACTION ROUTINES

409  
410

000001

.END

ERROR MESSAGES

.SBTTL ERROR MESSAGES

;; ERROR MESSAGES

92  
93  
94  
95  
96  
97  
98  
99  
100  
101

000004  
000046  
000106

.ENABL LC  
NTLERS ,2J,8B,CERR,RTSPC,CFLIN,<maximum retransmit count>  
NTLERS ,2K,8B,CERR,RTSPC,CFLIN,<retransmit timer value>  
NTLERS ,2L,8B,CERR,RTSPC,CFLIN,<holdback timer value>  
.DSABL LC

```

112      ;
113      ; -      WILL EVENTUALLY CRASH THE SYSTEM.
114
115      .MCALL      wdf$
116
117      000000      NHWDF$      ; DEFINE HARDWARE OFFSETS
118
119      000000      KS6:      .BLKW      1      ; LOCAL LOCATION OF KISAR6 ADDRESS
120      ;
121      ; SINCE WE WILL BE UNMAPPING OUR APR6
122      ; WE CANNOT ACCESS VECTOR TABLE

```

```

127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183

```

```

+
$ALN18 - ALLOCATE DOUBLEWORD ADDRESSABLE STORAGE FROM THE NON-UMR
          PORTION OF THE NETWORK BUFFER POOL
$ALL18 - ALLOCATE DOUBLEWORD ADDRESSABLE STORAGE FROM THE UMR
          PORTION OF THE NETWORK BUFFER POOL

INPUTS:
R1      = NUMBER OF BLOCKS (MAPPED) OR BYTES (UNMAPPED) NEEDED
$NBIAS  = BIAS OF BASE OF NEXT FREE NON-UMR SEGMENT (0=NONE)
$QBIAS  = BIAS OF BASE OF NEXT FREE UMR SEGMENT (0=NONE)

OUTPUTS:
C-BIT   = SUCCESS/FAILURE
R0      = BIAS OF BASE OF BUFFER
R1      = DESTROYED (110/IAS ONLY)
$NBIAS  = BIAS OF NEXT FREE NON-UMR SEGMENT
$QBIAS  = BIAS OF NEXT FREE UMR SEGMENT

NOTE:
THESE ROUTINES ALLOCATE ON 32-WORD BOUNDARIES, AND THUS
THE BUFFER VIRTUAL ADDRESS LOW THIRTEEN BITS WILL ALWAYS BE
ZERO. IF THE CALLER INTENDS TO RETURN THE MEMORY ALLOCATED
VIA THE $DEA18 ROUTINE, THE BUFFER SIZE (AS CALLED) MUST
BE PRESERVED. THE GLOBAL VARIABLES $QBIAS AND $NBIAS MUST
BE PROPERLY INITIALIZED BEFORE THIS ROUTINE IS CALLED. SEE
COMMENTS IN THE $AMEM ROUTINE ( MODULE 'ALCMEM' ).

-

.ENABL  LSB

$ALN18::.IF  DF,R,$$11M & M$$MGE      :IF NON-11M, NON-MGE, ONLY USE $ALL18
          SWSTK$ 30$                  :ENTER KERNEL MODE
          MOV    R1,R5                  :COPY NUMBER OF BLOCKS FOR $AMEM
          BEQ    10$                    :BR IF NONE REQUIRED, BAD CALL
          MOV    @NBIAS,R4              :GET BIAS OF FIRST NON-UMR SEGMENT
          CALL   $AMEM                  :TRY TO ALLOCATE SOME MEMORY
          BCS    5$                     :BR IF CAN'T, TRY UMR SEGMENT
          MOV    R4,@NBIAS              :RESET $NBIAS VALUE TO NEW SEGMENT
          BR     25$                    :GO SET USER R0 AND LEAVE
          .ENDC

$ALL18::
          SWSTK$ 30$                  :ENTER KERNEL MODE
          MOV    R1,R5                  :COPY BLOCK COUNT
          BEQ    10$                    :IF ZERO, RETURN FAILURE
          MOV    @QBIAS,R4              :GET BIAS OF FIRST FREE SEGMENT
          CALL   $AMEM                  :TRY TO ALLOCATE FROM NETWORK POOL
          BCC    20$                    :IF CC, SUCCESS
          MOV    @HEADR,R0              :GET USER MODE SP
          MOV    (R0),R0                :
          SET    USER MODE C-BIT
          LEAVE  R0 UNCHANGED
          SET    NEW $QBIAS VALUE
          MOV    R4,@QBIAS              :RETURN BUFFER BIAS IN R0
          MOV    R5,2(CSP)              :BACK TO USER MODE AND THEN TO CALLER
          RETURN

.DSABL  LSB

```

```

223 000272 001413      BEQ 101$      ; IF ZERO, ILLEGAL
224 000274      ASL$ 5,R0      ; CONVERT TO PSW PRIORITY VALUE
225 000306 010067 000000G    MOV R0,$$PRI ; STORE IT
226 000312 042767 000000G 000000G BIC #MS.PRI,$MISS ; IT'S NO LONGER MISSING
227 000320      10$: RETURN
228 000322      101$: MSG$R 1F      ; ILLEGAL INTERRUPT PRIORITY VALUE
229
230      ;
231      ; UNIBUS RUN MASK (CNT$DF)
232      ;
233      ; IF DF R$$MPL
234 C.URM: MOV .PNUMB,$URM      ; STORE THE UNIBUS RUN MASK
235      TSTB @NCPU      ; IS THIS A MULTI-PROCESSOR SYSTEM ?
236      BEQ 10$      ; IF EQ, NO
237      MOV $$CSR,R0      ; GET THE CSR ADDRESS
238      CALL $NLCSR      ; CHECK IF DEVICE IS OFFLINE
239      BCS 101$      ; IF CS, OFFLINE
240      10$: RETURN
241
242      101$: MSG$R 22      ; DEVICE OFFLINE
243
244      ;
245      ; CHECK IF UNIBUS RUN MASK IS MISSING
246      ;
247      ; MP.URM:
248      TSTB @NCPU      ; IS THIS A MULTI-PROCESSOR SYSTEM ?
249      BEQ 10$      ; IF EQ, NO
250      TST URMFL      ; IS THE UNIBUS RUN MASK PRESENT ?
251      BEQ 101$      ; IF EQ, NO
252      10$: RETURN
253
254      101$: MSG$R 2K      ; UNIBUS RUN MASK MISSING
255
256      ;
257      ; RE-CONFIGURATION DEVICE
258      ;
259      C.RCD1: MOVB .PCHAR,$RCDEV ; STORE THE 1ST CHAR OF THE RE-CONFIGURATION DEV
260      RETURN
261      C.RCD2: MOVB .PCHAR,$RCDEV+1 ; STORE THE 2ND CHAR OF THE RE-CONFIGURATION DEV
262      RETURN
263      .ENDC
264
265      000001      .END
266

```

```

A$$CHK= 000000      D$$SYNC= 000000      L$$ASG= 000000      P$$P45= 000000      $DNUMB= 000014
A$$CPS= 000000      D$$SYNM= 000000      L$$DRV= 000000      P$$WRD= 000000      $FOS = 000012
A$$PRI= 000000      D.EXT= 000052R      L$$P11= 000001      P.INC = ***** GX  $ERRIT= ***** GX
A$$TRP= 000000      D.NAME= ***** GX  L$$11R= 000000      Q$$OPT= 000010      $EXIT = 000000
BITS 000064R      002 END= 000154R      002 MS.EXT= ***** GX  R$$DER= 000000      $FAIL = 177777
BIT 1 000114R      002 E$$XPR= 000000      M$$CRB= 000124      R$$K11= 000001      $LAMDA= 000000
BIT 2 000130R      002 FG.X2P= ***** GX  M$$CRX= 000000      R$$SND= 000000      $MISS = ***** GX
CFGFB= ***** GX  FGLVL= 000001      M$$FCF= 000000      R$$T1M= 000000      $NUMBR= 000002
CFGFLG= ***** GX  G$$TPP= 000000      M$$MGE= 000000      SYNERR= ***** GX  $QDDM 000000RG
CFGSZ = ***** GX  G$$TSS= 000000      M$$NET= 000000      S$$WRG= 000000      $RAD50= 000016
C$$CKP= 000000      G$$TTK= 000000      M$$OVR= 000000      S$$YSZ= 007600      $STRNG= 000004
C$$ORE= 000400      G$$WRD= 000000      N$$ACC= 000001      T$$KMG= 000000      $SUBXP= 000010
C$$RSH= 177564      I$$RAR= 000000      N$$BUF= 000001      T$$MIN= 000000      $EXT = ***** GX
DDMDF 000000R      002 I$$RDN= 000000      N$$LDV= 000001      V$$CTR= 001000      $$$FLG= 177777
DDMKW 000000RG      003 I$$CNT= 177546      N$$MCP= 000001      X$$DBT= 000000      $$$KEY= 000005
DDMNAM 000016R      002 K$$CSR= 177546      N$$MLL= 000001      $ALPHA= 000022      $$$STA= 000000
DDMST 000000RG      002 K$$LDC= 000000      N$$MOV= 000010      $ANY = 000020      $$$TMP= 000043R 004
D$$BUG= 177514      K$$TPS= 000074      N$$NCT= 000001      $BLANK= 000006      .PNUMB= ***** GX
D$$ISK= 000000      LD$LP= 000000      N$$PEM= 000001      $DIGIT= 000024      .TPARS= ***** GX
D$$L11= 000001

```

. ABS. 000000 000 (RW,I,GBL,ABS,QVR)  
000072 001 (RW,I,LCL,REL,CON)  
\$STATE 000164 002 (RW,D,LCL,REL,CON)  
\$KTAB 000014 003 (RW,D,LCL,REL,CON)  
\$KSTR 000052 004 (RW,D,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 11343 Words ( 45 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:29.57

SY:CFGDDM.V2,[132,134]CFGDDM/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFGDDM



```

244          .SBTTL DTE$DF ACTION ROUTINES
245
246          ; START OF DTE ADDRESS
247
248 000052 005067 000000G DTEAST: CLR DTEFLG ; INITIALISE FLAGS (ON/OFF BITS)
249 000056 012700 000017 MOV #DTEALN,R0 ; GET LENGTH OF DTE ADDRESS
250 000062 012701 000000' MOV #DTEADD,R1 ; GET ADDRESS OF BUFFER
251 000066 105021 10$: CLR (R1)+ ; INITIALIZE BUFFER
252 000070 005300 DEC R0 ; MORE TO INITIALIZE?
253 000072 003375 BGT 10$ ; BR IF YES
254 000074 105067 000035' CLR COUNT ; INITIALIZE DTE DIGIT COUNT
255 000100 012767 000000' 000040' MOV #DTEADD,NEXT ; START STORING AT BEGINNING OF BUFFER
256 000106 CALL STRNXT ; STORE CHARACTER
257 000112 RETURN
258
259          ; PROCESS DTE ADDRESS DIGIT
260
261
262 000114 026727 000040' 000016' DTEDIG: CMP NEXT,#DTEND ; IS ADDRESS TOO MANY CHARACTERS?
263 000122 101003 BHI 101$ ; BR IF YES
264 000124 CALL STRNXT ; STORE NEXT DIGIT
265 000130 RETURN
266
267          ; ERRORS
268
269 000132 101$: MSG$R 03 ; ILLEGAL DTE ADDRESS
270
271          ; END OF DTE ADDRESS
272
273
274 000140 004567 000000G DTEAEN: JSR R5,$$AVRG ; SAVE R3-R5
275 000144 012700 000010 MOV #<DTEALN+1>/2,R0 ; GET LENGTH OF PACKED ADDRESS
276 000150 012704 000000' MOV #DTEADD,R4 ; GET ADDRESS OF DTE ADDRESS BUFFER
277 000154 012705 000017' MOV #DTEPCK,R5 ; GET ADDRESS OF PACKED ADDRESS BUFFER
278 000160 CALL PCKBCD ; PACK ADDRESS IN BCD FORMAT
279
280 000164 017700 000000G MOV @PSIPT,R0 ; POINT TO HOME BLOCK
281 000170 062700 000024 ADD #H$NETW,R0 ; POINT TO DEFAULT NET NAME
282 000174 012701 000027' MOV #DTNTNM,R1 ; POINT TO LOCAL STORAGE OF NAME
283 000200 012702 000006 MOV #DTNMMX,R2 ; GET SIZE OF NAME
284 000204 112021 10$: MOV (R0)+,(R1)+ ; SET UP DEFAULT FOR DTE
285 000206 005302 DEC R2 ; MORE TO SET UP?
286 000210 003375 BGT 10$ ; YES, SET AWAY
287 000212 RETURN ; ELSE COMPLETED

```

CFGDTE CREATED BY MACRO ON 29-JUN-85 AT 00:03 PAGE 3 M 6  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL    | VALUE      | REFERENCES                                      |
|-----------|------------|---|
| SF.MFL    | = 000040   | #5-66   |
| SF.PAC    | = 000020   | #5-66   |
| SF.REA    | = 000010   | #5-66   |
| SF.SER    | = 000001   | #5-66   |
| SF.SVC    | = 000002   | #5-66   |
| SF.UNL    | = 000040   | #5-66   |
| SPACE     | = 000040   | #6-83   |
| STRNXT    | = 000610 R | 12-256 12-264 #19-462                           |
| SYNERR    | = ***** GX | *9-157 *9-160                                   |
| \$\$\$BAS | = *****    | 8-129 8-130 8-130 8-131 8-131 8-132 8-132 8-133 |
| S.COST    | 000001     | #5-66   |
| S.FLG     | 000000     | #5-66   |
| S.LEN     | 000004     | #5-66   |
| S.NMST    | 000002     | #5-66   |
| S.OWNR    | 000003     | #5-66   |
| \$ALPHA   | = 000022   | #10-173   |
| \$ANY     | = 000020   | #10-173   |
| \$BLANK   | = 000006   | #10-173   |
| \$CAT5    | = ***** GX | 18-424  |
| \$CDTB    | = ***** GX | 19-467  |
| \$CEACX   | = ***** GX | 16-379  |
| \$DIGIT   | = 000024   | #10-173   |
| \$DNUMB   | = 000014   | #10-173   |
| \$EOS     | = 000012   | #10-173   |
| \$ERROO   | 000046 R   | #8-129 14-338                                   |
| \$ERRO?   | 000106 R   | #8-130 12-269 15-350                            |
| \$ERRJ4   | 000140 R   | #8-131 13-307                                   |
| \$ERRO5   | 000170 R   | #8-132 13-318                                   |
| \$ERRO6   | 000224 R   | #8-133 18-444                                   |
| \$ERRIT   | = ***** GX | 9-167   |
| \$EXIT    | = 000000   | #10-173   |
| \$FAIL    | = 177777   | #10-173   |
| \$GPRM    | = *****    | 10-173  |
| \$HEADR   | = ***** GX | 16-390  |
| \$LAMDA   | = 000000   | #10-173   |
| \$NUMBR   | = 000002   | #10-173   |
| \$ODTE    | 000000 RG  | #9-152  |
| \$RAD50   | = 000016   | #10-173   |
| \$RONLY   | = *****    | 10-173 10-173                                   |
| \$SAVRG   | = ***** GX | 12-274  |
| \$STRNG   | = 000004   | #10-173   |
| \$SUBXP   | = 000010   | #10-173   |
| \$\$\$FLG | = 177777   | #10-173   |
| \$\$\$KEY | = 177777   | #10-173   |
| .PCHAR    | = ***** GX | 19-462  |
| .PNUMB    | = ***** GX | 13-294 18-432                                   |
| .PNUMH    | = ***** GX | 13-292 18-430 18-437                            |
| .PSTCN    | = ***** GX | 14-325 18-421                                   |
| .PSTPT    | = ***** GX | 14-328 18-423                                   |
| .TPARS    | = ***** GX | 9-158   |

CFGPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 <sup>N 7</sup> 00:04  
Table of contents

|     |     |   |
|-----|-----|---|
| 5-  | 54  | MACRO DEFINITIONS                       |
| 6-  | 97  | LOCAL SYMBOL DEFINITIONS                |
| 7-  | 115 | LOCAL DATA                              |
| 8-  | 157 | ERROR MESSAGES                          |
| 9-  | 174 | LOOK FOR PVC\$DF MACRO                  |
| 10- | 204 | TPARS STATE TABLES                      |
| 11- | 293 | PVC\$DF ACTION ROUTINES                 |
| 14- | 467 | ALLXCB - ALLOCATE X25 CIRCUIT BLOCK     |
| 15- | 507 | CHKPVC - CHECK FOR VALID PVC            |
| 16- | 548 | FNDPRT - FIND FREE PORT ENTRY           |
| 17- | 586 | INIXCB - INITIALIZE X25 CIRCUIT BLOCK   |
| 18- | 630 | FNDHSH - FIND HASH TABLE ENTRY          |
| 19- | 678 | BIASX - BIAS XPOOL ADDRESS FOR APR6     |
| 20- | 698 | PVDLM - CHECK FOR DLM PVC               |
| 21- | 759 | DLMTLC - SET UP TRANSPORT LINE COUNTERS |

```

467 .SBTTL ALLXCB - ALLOCATE X25 CIRCUIT BLOCK
468
469
470
471
472
473
474
475
476
477
478
479
480
481 ALLXCB: MOV #X$LEN,R1 ;; GET SIZE OF X.25 CIRCUIT BLOCK
482 CALL $XALOC ;; ALLOCATE CIRCUIT BLOCK
483 BCS 50$ ;; BR IF UNSUCCESSFUL ALLOCATION
484 MOV @PSIPT,R2 ;; POINT TO PSI HOME BLOCK
485 MOV H$PTB(R2),-(SP) ;; GET UNMAPPED ADDRESS OF PORT TABLE
486 CALL $CEACX ;; MAP TO PORT TABLE
487 TST (SP)+ ;; CLEAN UP STACK
488 MOV R0,@PRTADD ;; STORE UNMAPPED ADDRESS IN PORT TABLE
489 CALL INIXCB ;; INITIALIZE X25 CIRCUIT BLOCK
490 CALL FNDHSH ;; FIND HASH TABLE ENTRY AND SET UP
491 BCS 50$ ;; BR IF NO ENTRY AVAILABLE
492 MOV PVCOWN,R0 ;; GET PROCESS OWNER NAME
493 BEQ 40$ ;; IF EQ, NONE SPECIFIED
494 MOV DLM$LN,R1 ;; GET DLM'S SYSTEM LINE NUMBER
495 BIS #200,R1 ;; INDICATE SYSTEM LINE NUMBER PRESENT
496 BIT #PF.DLM,FLAG ;; IS THIS A DLM CIRCUIT?
497 BNE 10$ ;; IF NE, YES
498 CALL FNDPDV ;; GET PDV INDEX
499 MOV R1,X$MOWN(R3) ;; STORE PDV INDEX
500 MOV ZTIME,-(SP) ;; GET $ZTIME ADDRESS
501 ADD #2,(SP) ;; GET $ZTIME+2 ADDRESS
502 MOV (SP),X$TCLZ(R3) ;; STORE TIME SINCE LAST ZEROED
503 TST (SP)+ ;; CLEANUP STACK
504
505 10$: MOV R1,X$MOWN(R3)
506 40$: MOV ZTIME,-(SP)
507 50$: RETURN

```

| SYMBOL  | VALUE      | REFERENCES |         |         |         |        |        |             |
|---------|------------|------------|---------|---------|---------|--------|--------|-------------|
| ALLXCB  | 000724 R   | 13-448     | #14-481 |         |         |        |        |             |
| BIASX   | 001610 R   | 17-611     | #19-691 |         |         |        |        |             |
| CERR    | = ***** GX | 8-161      | 8-162   | 8-163   | 8-164   | 8-165  | 8-166  | 8-167 8-168 |
| CFGBF   | = ***** GX | 9-191      |         |         |         |        |        |             |
| CFGSZ   | = ***** GX | 9-190      |         |         |         |        |        |             |
| CFLIN   | = ***** GX | 8-161      | 8-162   | 8-163   | 8-164   | 8-165  | 8-166  | 8-167 8-168 |
| CHKBLK  | = ***** GX | 12-353     |         |         |         |        |        |             |
| CHKPVC  | 001050 R   | 13-413     | #15-522 |         |         |        |        |             |
| CHKWND  | = ***** GX | 12-365     |         |         |         |        |        |             |
| CHNLMX  | = ***** GX | 11-328     |         |         |         |        |        |             |
| CSDTE   | 000012     | *13-438    | 15-534  |         |         |        |        |             |
| C\$FLG  | 000015     | *13-436    |         |         |         |        |        |             |
| C\$LCN  | 000010     | *13-437    | 15-532  |         |         |        |        |             |
| C\$LEN  | 000016     | 13-415     |         |         |         |        |        |             |
| C\$NAM  | 000002     | 13-430     | 15-539  |         |         |        |        |             |
| C\$PORT | 000014     | *13-447    |         |         |         |        |        |             |
| DLMCTL  | 000044 R   | *7-146     | *20-737 |         |         |        |        |             |
| DLMIND  | 000046 R   | *7-148     | *20-722 |         |         |        |        |             |
| DLMPDV  | 000022 R   | *7-137     | *20-724 |         |         |        |        |             |
| DLM\$LN | 000030 R   | *7-140     | 14-494  | *21-780 | *21-781 |        |        |             |
| DLM\$LT | 000024 R   | *7-138     |         |         |         |        |        |             |
| DLM\$TA | 000026 R   | *7-139     | 13-407  | *20-749 | 21-785  |        |        |             |
| DLM\$TL | 000034 R   | 13-445     | #21-775 |         |         |        |        |             |
| DLMUNT  | 000045 R   | *7-147     |         |         |         |        |        |             |
| DTEDES  | = ***** GX | 13-438     | 15-534  | 17-615  | 18-647  |        |        |             |
| E\$NBR  | 000014     | #5-66      |         |         |         |        |        |             |
| E\$NBS  | 000020     | #5-66      |         |         |         |        |        |             |
| E\$NCR  | 000034     | #5-66      |         |         |         |        |        |             |
| E\$NCS  | 000036     | #5-66      |         |         |         |        |        |             |
| E\$NIC  | 000044     | #5-66      |         |         |         |        |        |             |
| E\$NLEN | 000050     | #5-66      |         |         |         |        |        |             |
| E\$NLLA | 000012     | #5-66      |         |         |         |        |        |             |
| E\$NLNK | 000000     | #5-66      |         |         |         |        |        |             |
| E\$NML  | 000040     | #5-66      |         |         |         |        |        |             |
| E\$NMR  | 000024     | #5-66      |         |         |         |        |        |             |
| E\$NMS  | 000030     | #5-66      |         |         |         |        |        |             |
| E\$NNOD | 000002     | #5-66      |         |         |         |        |        |             |
| E\$NRT  | 000042     | #5-66      |         |         |         |        |        |             |
| E\$NRTP | 000005     | #5-66      |         |         |         |        |        |             |
| E\$NSEG | 000010     | #5-66      |         |         |         |        |        |             |
| E\$NTIM | 000046     | #5-66      |         |         |         |        |        |             |
| E\$NUSE | 000004     | #5-66      |         |         |         |        |        |             |
| E\$STRT | 000006     | #5-66      |         |         |         |        |        |             |
| FLAG    | 000032 R   | *7-141     | *9-186  | 13-411  | 13-419  | 13-443 | 13-451 | 14-496      |
| FMT8    | = ***** GX | 8-164      |         |         |         |        |        |             |
| FMT8B   | = ***** GX | 8-161      | 8-162   | 8-163   | 8-165   | 8-166  | 8-167  | 8-168       |
| FM.8    | = 000000   | #8-164     |         |         |         |        |        |             |
| FM.8B   | = 000000   | #8-161     | #8-162  | #8-163  | #8-165  | #8-166 | #8-167 | #8-168      |
| FNDHSH  | 001452 R   | 14-490     | #18-646 |         |         |        |        |             |
| FNDPDV  | = ***** GX | 14-498     |         |         |         |        |        |             |
| FNDPRT  | 001172 R   | 13-439     | #16-565 |         |         |        |        |             |
| HSHADD  | = ***** GX | *12-655    | 18-664  |         |         |        |        |             |

CFGSLT - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:05 Page 7-1  
TPARS TABLES

160 000062  
161  
162

TRANS \$NUMBR,\$EXIT,\$RPRI

```

82          .SBTTL $QSTA - LOOK FOR STA$DF
83
84          ;+
85          $QSTA - LOOK FOR STA$DF
86          :
87          INPUTS:
88          :   NONE
89          :
90          OUTPUTS:
91          :   C-BIT - SUCCESS/FAILURE
92          :
93          :
94          :-
95
96 000216 012705 000000' $QSTA:: MOV    #STADF,R5 ; STATE TABLE ADDRESS
97 000222 005001          CLR    R1      ; FULL KEYWORD MATCH LENGTH
98 000224 012702 000000'      MOV    #STAKW,R2 ; KEYWORD TABLE ADDRESS
99 000230 016703 000000G      MOV    CFGSZ,R3 ; RECORD LENGTH
100 000234 012704 000000G      MOV    #CFGBF,R4 ; RECORD BUFFER ADDRESS
101 000240 005067 000000G      CLR    SYNERR ; CLEAR SYNTAX ERROR FLAG
102 000244          CALL    TPARS ; GO DO THE PARSE
103 000250 103003          BCC    20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
104 000252 005367 000000G      DEC    SYNERR ; DID SYNTAX ERROR OCCUR?
105 000256 001401          BEQ    101$ ; IF EQ, YES
106 000260          20$: RETURN
107
108          : ERRORS
109
110 000262          101$: MSG$R 1T ; SYNTAX ERROR

```

```

54      .SBTTL  MACRO DEFINITIONS
55
56      ;
57      ; LIBRARY MACROS
58      ;
59      .MCALL  ISTAT$,TRANS$,STAT$,EMSG$,NTL$,XPDD$,CTRDF$
60      .MCALL  PHBDF$,RETC$,SVCDF$,ASL$,SLTDF$,PLBDF$
61
62      CTRDF$      ; DEFINE TRANSPORT LINE COUNTER OFFSETS
63      XPDD$      ; DEFINE XPT DDB OFFSETS
64      PHBDF$      ; DEFINE PSI HOME BLOCK OFFSETS
65      SLTDF$      ; DEFINE SLT OFFSETS
66      SVCDF$      ; DEFINE SVC OFFSETS
67      PLBDF$      ; DEFINE PLB OFFSETS
68
69      ;
70      ; LOCAL MACRO DEFINITIONS
71      ;
72      ;
73      ; REJECT TPARS TRANSITION
74      ;
75      .MACRO  REJ$
76      ADD    #2,(SP)      ; RETURN TO CALLER+2
77      CLR    SYNERR      ; INDICATE NO SYNTAX ERROR
78      .ENDM  REJ$
79
80      ;
81      ; SAVE CURRENT MAPPING
82      ;
83      .MACRO  SAVMAP
84      MOV    @KSAR5,-(SP) ; SAVE APR 5
85      .ENDM
86
87      ;
88      ; RESTORE PREVIOUS MAPPING
89      ;
90      .MACRO  RESMAP
91      MOV    (SP)+,@KSAR5 ; RESTORE APR5
92      .ENDM
93
94      ;
95      ; SET UP APR5 MAPPING
96      ;
97      .MACRO  MAP    BIAS
98      MOV    BIAS,@KSAR5 ; SET UP APR5 MAPPING
99      .ENDM
100
101      ;
102      ; SET UP APR5 BIAS
103      ;
104      .MACRO  BIAS    REG
105      BIC    #16000,REG ; CLEAR CURRENT MAPPING BITS
106      BIS    #12000,REG ; SET UP APR5 MAPPING
107      .ENDM

```



M 13

```

SETTLC - SET UP TRANSPORT LINE COUNTERS

576 001376 042701 160000      BIC    #160000,R1      ;; CLEAR CURRENT APR MAPPING BITS
577 001402 052701 140000      BIS    #140000,R1      ;; XPT NEEDS APR6 MAPPING
578 001406 010062 000034      MOV     R0,T$SVC(R2)    ;; STORE APR BIAS OF SVC DESCRIPTOR
579 001412 010162 000036      50$:  MOV     R1,T$SVC+2(R2)  ;; STORE VIRTUAL ADDR OF SVC DESCRIPTOR
580 001416 016762 000002* 000030  MOV     HTIMR,T$T5(R2)  ;; STORE HELLO TIMER
581 001424 016762 000004* 000032  MOV     LTIMR,T$T6(R2)  ;; AND LISTEN TIMER
582 001432      RESMAP      ;; RESTORE PREVIOUS MAPPING
583 001436      RETURN

```

N 13

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51

.TITLE CFGUNT - CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 09-JUN-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

A$$CHK= 000000
A$$CPS= 000000
A$$PRI= 000000
A$$RP= 000000
CERR = ***** GX
CFGBF = ***** GX
CFGFSZ = ***** GX
CFLIN = ***** GX
CHERR 000254R
CH.DUP= 000001
CH.D11= 000100
CH.ECH= 000002
CH.MDT= 000020
CH.PAR= 000100
CH.PRM= 000200
CH.PRT= 000007
CH.STA= 000040
CH.SYN= 000004
CH.TYP= 000010
COST 000052R
COST1 000062R
C$$CKP= 000000
C$$ORE= 000400
C$$RSH= 177564
C1.BSY= 000002
C1.DCP= 000001
C1.SDL= 000003
C1.X25= 000004
DPR 000072R
DPR1 000102R
D$$BUG= 177514
D$$ISK= 000000
D$$L11= 000001
D$$YNC= 000000
D$$YNM= 000000
END 000130R
E$$XPR= 000000
FL.FDX= ***** GX
FL.HDX= ***** GX
FL.KMX= ***** GX
FL.LMC= ***** GX
FMTB = ***** GX
FMT8A = ***** GX

FMT8B = ***** GX
FM.8 = 000000
FM.8A = 000000
FM.8B = 000000
F$$LVL= 000001
GETCSR= ***** GX
G$$TTP= 000000
G$$TSS= 000000
G$$TTK= 000000
G$$WRD= 000000
I$$RAR= 000000
I$$RDN= 000000
K$$CNT= 177546
K$$CSR= 177546
K$$LDC= 000000
K$$TPS= 000074
LD$LP = 000000
LF.ACT= 100000
LF.BRO= 000400
LF.BWT= 000007
LF.ENA= 002000
LF.LPB= 001000
LF.MDC= 000100
LF.MFL= 004000
LF.MTP= 000020
LF.PAC= 000200
LF.RDY= 040000
LF.REA= 010000
LF.SER= 000040
LF.T1M= 000310
LF.UNL= 020000
LF.X2P= 000000
LN.CLO= 000000
LN.DUM= 000005
LN.LOA= 000004
LN.LOO= 000003
LN.OAU= 000003
LN.OFF= 000001
LN.ON = 000000
LN.OOP= 000004
LN.OPE= 000001
LN.REF= 000002
LN.SER= 000002

LN.STA= 000017
LN.SUB= 000360
LN.TRI= 000006
L$$ASG= 000000
L$$DRV= 000000
L$$P11= 000001
L$$11R= 000000
L.COST 000015
L.CTL 000012
L.CVA 177776
L.DDM 000002
L.DDS 000004
L.DLC 000003
L.DLM 000006
L.DLS 000010
L.FLG 000000
L.KRBA 000016
L.LEN = 000022
L.MPF 000022
L.NMST 000020
L.NSTA 000014
L.OWNR 000021
L.UNT 000013
MAXCST= 000031
MS.SEC= ***** GX
M$$CRB= 000124
M$$CRX= 000000
M$$FCS= 000000
M$$MGE= 000000
M$$NET= 000000
M$$OVR= 000000
N$$ACC= 000001
N$$BUF= 000001
N$$LDV= 000001
N$$MCP= 000001
N$$MLL= 000001
N$$MOV= 000010
N$$NCT= 000001
N$$PEM= 000001
PECH 000112R
PECH1 000122R
P$$P45= 000000

P$$WRD= 000000
Q$$OPT= 000010
RDBSZ = ***** GX
REP8A = ***** GX
RTSPC = ***** GX
R$$DER= 000000
R$$K11= 000001
R$$SND= 000000
R$$11M= 000000
SETTMO 000400R
SF.ACT= 000200
SF.ENA= 000100
SF.LPB= 000004
SF.MFL= 000040
SF.PAC= 000020
SF.REA= 000010
SF.SER= 000001
SF.SVC= 000002
SF.UNL= 000040
SPEEDT 000000R
SYNERR= ***** GX
S$$WRG= 000000
S$$YST= 007600
S.COST 000001
S.FLG 000000
S.LEN 000004
S.NMST 000002
S.OWNR 000003
T$$KMG= 000000
T$$MIN= 000000
UCNT = ***** GX
UNTDf 000000R
UNTKW 000000RG
UNTST 000000RG
U.CHAO 000156R
U.CHAI 000262R
U.CST 001022R
U.DPR 001114R
U.PECH 000334R
U.SCSR 000614R
U.UNT 000112R
U.XCSR 000756R

V$$CTR= 001000
X$$DBT= 000000
$ALPHA= 000022
$ANY = 000020
$BLANK= 000006
$DIGIT= 000024
$DIV = ***** GX
$DNUMB= 000014
$EOS = 000012
$ERRYP= ***** GX
$ERRZO 000254R
$ERR1G 000000R
$ERR1H 000024R
$ERR1K 000060R
$ERR1L 000116R
$ERR1M 000154R
$ERR1N 000216R
$ERRIT= ***** GX
$EXIT = 000000
$FAIL = 177777
$FLAGS= ***** GX
$LAMDA= 000000
$MISS = ***** GX
$NLCSR= ***** GX
$NUMBER= 000002
$QUNT 000040RG
$RAD50= 000016
$SLTA = ***** GX
$STRNG= 000004
$SUBXP= 000010
$SDCHA= ***** GX
$SDPR = ***** GX
$SPCHA= ***** GX
$SCSR= ***** GX
$SFLG= 177777
$SKEY= 000000
$SSTA= 000000
$STMP= 000000R
.PNUMB= ***** GX
.PNUMH= ***** GX
.TPARS= ***** GX
..SCSR= ***** GX

000 (RW,I,GBL,ABS,OVR)
001 (RW,I,LCL,REL,CON)
002 (RW,D,LCL,REL,CON)
003 (RW,D,LCL,REL,CON)
004 (RW,D,LCL,REL,CON)
005 (RW,D,LCL,REL,CON)
Errors detected: 0

```

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0

```

103                                     .SBTTL $QX2P - LOOK FOR X2P$DF MACRO
104
105                                     ;+
106                                     ; $QX2P - LOOK FOR X2P$DF MACRO
107                                     ;
108                                     ; INPUTS:
109                                     ;     NONE
110                                     ;
111                                     ; OUTPUTS:
112                                     ;     ALL REGISTERS DESTROYED
113                                     ;
114                                     ; -
115                                     ;
116 000144 012705 000000' $QX2P:: MOV     #X2PDF,R5      ; STATE TABLE ADDRESS
117 000150 005001          CLR     R1                ; FULL KEYWORD MATCH LENGTH
118 000152 012702 000000' MOV     #X2PKW,R2          ; KEYWORD TABLE ADDRESS
119 000156 016703 000000G MOV     CFGSZ,R3            ; RECORD LENGTH
120 000162 012704 000000G MOV     #CFGBF,R4          ; RECORD BUFFER ADDRESS
121 000166 005067 000000G CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
122 000172          CALL    2PARS          ; GO DO THE PARSE
123 000176 103003          BCC     20$           ; IF CC, FOUND WHAT WE WERE LOOKING FOR
124 000200 005367 000000G DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
125 000204 001401          BEQ     101$          ; IF EQ, YES
126 000206          20$:   RETURN
127
128                                     ; ERRORS
129
130 000210          101$:   MSG$R 1T              ; SYNTAX ERROR

```

.SBTTL \$AMEM - ALLOCATE MEMORY IN NTPool

\*\$AMEM - SUBROUTINE TO ALLOCATE A CONTIGUOUS BLOCK OF MEMORY FROM PARTITION.

INPUTS:

R4 - CURRENT VALUE OF \$QBIAS  
 R5 - NUMBER OF 32 WORD BLOCKS REQUIRED  
 TASK SWITCHING MUST BE DISABLED

OUTPUTS:

C BIT SET IF NO MEMORY ALLOCATED  
 CLEAR IF MEMORY ALLOCATED  
 R4 - NEW VALUE OF \$QBIAS  
 P5 - BIT OF MEMORY ALLOCATED, IF ALLOCATION SUCCEEDS  
 R0,R1,R2 ARE DESTROYED

NOTE:

THE VALUE OF \$QBIAS RETURNED IN R4 MUST BE UPDATED IN ITS GLOBAL STORAGE LOCATION \$QBIAS BEFORE TASK-SWITCHING IS REENABLED.

REGISTER USAGE:

R0 - BUFFER FOR ADJUSTED HOLE SIZE  
 R1 - BIAS OF PREVIOUS HOLE  
 R4 - BUFFER FOR BIAS OF NEXT HOLE  
 R5 - SIZE (IN 32-WD BLKS) OF SPACE REQUIRED

\$AMEM::

```

158 000002
159
160 000002 016767 000000G 177770      MOV    KSAR6,KS6      ; COPY KISAR6 ADDRESS FROM VECTOR TABLE
161 000010 017746 177764              MOV    @KS6,-(SP)      ; SAVE MAPPING REGISTER
162
163
164 ; SPECIAL CASE -- IS SPACE AVAILABLE IN FIRST HOLE
165
166 000014 010477 177760      MOV    R4,@KS6      ; SET VIRTUAL BIAS AT FIRST HOLE
167 000020 000261              SEC                      ; ASSUME NOT
168 000022 001473              BEQ    70$      ; IS THERE A FIRST HOLE?
169                               ; NO -- RETURN WITH CARRY SET
170
171 000024 016700 140J02'      MOV    ,JASEB+2,R0      ; SET R0 TO HOLE SIZE, ADJUSTED AS IF REQUIRED
172 000030 160500              SUB     R5,R0      ; SPACE HAS BEEN TAKEN FROM HOLE.
173                               ; IS SUFFICIENT SPACE AVAILABLE?
174 000C32 103422              BLO    40$      ; NO -- CHECK NEXT HOLE
175                               ; YES -- IS ANY SPACE LEFT IN HOLE?
176 000034 001414              BEQ    30$      ; NO -- ALLOCATE MEMORY & DELETE HOLE
177                               ; YES -- ALLOCATE MEMORY & ADJUST HOLE SIZE
178
179 000036 017746 177736      MOV    @KS6,-(SP)      ; SET 14TH FIRST ADR OF ALLOCATED SPACE ON STACK
  
```

```

185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205 000220
206 000220
207 000224 010046
208 000226 010003
209 000230 017704 000000G
210 000234 020077 000000G
211 000240 103002
212 000242 017704 000000G
213 000246 010105
214 000250
215 000254 022677 000000G
216 000260 103003
217 000262 010477 000000G
218 000266 000402
219 000270 010477 000000G
220 000274
221
222
223 000001

;+
; $DEA18 - DE-ALLOCATE DOUBLEWORD ADDRESSABLE STORAGE IN BUFFER POOL
;
; INPUTS:
; R0 = BIAS OF BASE OF BUFFER TO BE RETURNED
; R1 = NUMBER OF BLOCKS (MAPPED) OR BYTES (UNMAPPED) IN BUFFER
; $NBIAS= BIAS OF BASE OF NEXT FREE NON-UMR BUFFER
; $QBIAS = BIAS OF BASE OF NEXT FREE UMR BUFFER
; $QSTRT= BIAS OF BASE OF UMR BUFFER POOL
;
; OUTPUTS:
; $NBIAS= BIAS OF BASE OF NEXT FREE NON-UMR BUFFER
; $QBIAS = BIAS OF BASE OF NEXT FREE UMR BUFFER
; R0,R1,R2 - DESTROYED (11D/IAS ONLY)
;
; NOTE:
; IF IMPROPER VALUES ARE SUPPLIED TO THIS ROUTINE, IT WILL
; PROBABLY RESULT IN THE CORRUPTION OF THE OPERATING SYSTEM.
;
;--
$DEA18::
SWSTK$ 30$ ;ENTER KERNEL MODE
MOV R0,-(SP) ;SAVE BIAS OF BUFFER TO RETURN
MOV R0,R3 ;COPY BUFFER BIAS FOR $DMEM
MOV @QBIAS,R4 ;GET CURRENT $QBIAS VALUE
CMP R0,@QSTRT ;IS BUFFER IN THE UMR POOL?
BHS 10$ ;BR IF YES, USE THE $QBIAS VALUE
MOV @NBIAS,R4 ;USE THE $NBIAS (NON-UMR) VALUE
10$: MOV R1,R5 ;COPY BLOCK COUNT FOR $DMEM
CALL $DMEM ;DE-ALLOCATE BACK TO NETWORK POOL
CMP (SP)+,@QSTRT ;DID BUFFER GO TO UMR OR NON-UMR?
BHS 20$ ;BR IF WENT TO UMR PORTION
MOV R4,@NBIAS ;RESET $NBIAS (NON-UMR)
BR 30$ ;LEAVE KERNEL MODE
20$: MOV R4,@QBIAS ;RESET $QBIAS (UMR)
30$: RETURN ;RETURN TO USER MODE, THEN TO CALLER

.END

```

|                  |                      |                  |                  |                      |
|------------------|----------------------|------------------|------------------|----------------------|
| A\$CHK= 000000   | D\$SYNM= 000000      | L\$P11= 000001   | RTSPC = ***** GX | \$ERR1T= ***** GX    |
| A\$CPS= 000000   | E\$XPR= 000000       | L\$11R= 000000   | R\$DER= 000000   | \$EXIT = 000000      |
| A\$PRI= 000000   | FMT8 = ***** GX      | MS.CSR= ***** GX | R\$K11= 000001   | \$FAIL = 177777      |
| A\$TRP= 000000   | FMT8A = ***** GX     | MS.PRI= ***** GX | R\$SND= 000000   | \$LMDA= 000000       |
| CERR = ***** GX  | FMT8B = ***** GX     | MS.VCT= ***** GX | R\$11M= 000000   | \$MISS = ***** GX    |
| CFGBF = ***** GX | FM.8 = 000000        | M\$CRB= 000124   | SYNERR= ***** GX | \$MXVCT= ***** GX    |
| CFGSZ = ***** GX | FM.8A = 000000       | M\$CRX= 000000   | S\$WRG= 000000   | \$NLCSR= ***** GX    |
| CFLIN = ***** GX | FM.8B = 000000       | M\$FCS= 000000   | S\$YSZ= 007600   | \$NUMBR= 000002      |
| CNTDF 000000R    | 003 F\$LVL= 000001   | M\$MGE= 000000   | T\$KMG= 000000   | \$QCNT 000000RG      |
| CNTKW 000000RG   | 004 GETCSR= ***** GX | M\$NET= 000000   | T\$SMIN= 000000  | \$RAD50= 000016      |
| CNTST 000000RG   | 003 G\$TTP= 000000   | M\$OVR= 000000   | V\$CTR= 001000   | \$STRNG= 000004      |
| CTL = ***** GX   | G\$TSS= 000000       | N\$ACC= 000001   | X\$DBT= 000000   | \$SUBXP= 000010      |
| C\$CKP= 000000   | G\$TTK= 000000       | N\$BUF= 000001   | \$ALPHA= 000022  | \$CSR = ***** GX     |
| C\$ORE= 000400   | G\$WRD= 000000       | N\$LDV= 000001   | \$ANY = 000020   | \$PRI = ***** GX     |
| C\$RSH= 177564   | I\$RAR= 000000       | N\$MCP= 000001   | \$BLANK= 000006  | \$VECT= ***** GX     |
| C.CSR 000164R    | I\$RDN= 000000       | N\$MLL= 000001   | \$DIGIT= 000024  | \$FLG= 177777        |
| C.CTL 000052R    | K\$CNT= 177546       | N\$MOV= 000010   | \$DNUMB= 000014  | \$KEY= 000000        |
| C.PRI 000240R    | K\$CSR= 177546       | N\$NCT= 000001   | \$EOS = 000012   | \$STA= 000000        |
| C.VECT 000074R   | K\$LD= 000000        | N\$PEM= 000001   | \$ERRZ2 000150R  | 002 \$TMP= 000000R   |
| D\$BUG= 177514   | K\$TPS= 000074       | P\$P45= 000000   | \$ERR1C 000000R  | 002 .PNUMB= ***** GX |
| D\$ISK= 000000   | LD\$LP = 000000      | P\$WRD= 000000   | \$ERR1D 000032R  | 002 .PNUMH= ***** GX |
| D\$L11= 000001   | L\$A3G= 000000       | Q\$OPT= 000010   | \$ERR1E 000070R  | 002 .TPARS= ***** GX |
| D\$YNC= 000000   | L\$DRV= 000000       | REP8A = ***** GX | \$ERR1F 000114R  | 002                  |

  

|                |                        |
|----------------|------------------------|
| . ABS. 000000  | 000 (RW,I,GBL,ABS,OVR) |
| 000330         | 001 (RW,I,LCL,REL,CON) |
| DATA 000170    | 002 (RW,D,LCL,REL,CON) |
| \$STATE 000036 | 003 (RW,D,LCL,REL,CON) |
| \$XTAB 000002  | 004 (RW,D,LCL,REL,CON) |
| \$KSTR 000007  | 005 (RW,D,LCL,REL,CON) |

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 12504 Words ( 49 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLIIS

Elapsed time: 00:00:21.06

SY:CFGCNT.V2,[132,134]CFGCNT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFGCNT

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE      | REFERENCES  |
|----------|------------|-------------|
| CFGBF    | = ***** GX | 6-77        |
| CFGSZ    | = ***** GX | 6-76        |
| DDMDF    | 000000 R   | 6-73        |
| DDMKW    | 000000 RG  | 6-75 #7-92  |
| DDMST    | 000000 RG  | #7-92       |
| D.EXT    | 000052 R   | #8-157      |
| MS.EXT   | = ***** GX | 8-158       |
| M\$MGE   | = 000000   | 7-101 8-156 |
| SYNERR   | = ***** GX | *6-78 *6-81 |
| \$ALPHA  | = 000022   | #7-92       |
| \$ANY    | = 000020   | #7-92       |
| \$BLANK  | = 000006   | #7-92       |
| \$DIGIT  | = 000024   | #7-92       |
| \$DNUMB  | = 000014   | #7-92       |
| \$EOS    | = 000012   | #7-92       |
| \$ERRIT  | = ***** GX | 6-88        |
| \$EXIT   | = 000000   | #7-92       |
| \$FAIL   | = 177777   | #7-92       |
| \$GPRM   | = *****    | 7-92        |
| \$LAMD A | = 000000   | #7-92       |
| \$MISS   | = ***** GX | 8-158       |
| \$NUMBR  | = 000002   | #7-92       |
| \$QDDM   | 000000 RG  | #6-73       |
| \$RAD50  | = 000016   | #7-92       |
| \$RCNLY  | = *****    | 7-92 7-92   |
| \$STRNG  | = 000004   | #7-92       |
| \$SUBXP  | = 000010   | #7-92       |
| \$SEXT   | = ***** GX | *8-160      |
| \$SFLG   | = 177777   | #7-92       |
| \$SKEY   | = 177777   | #7-92       |
| .PNUMB   | = ***** GX | 8-160       |
| .TPARS   | = ***** GX | 6-79        |



```

289      ;
290      ; HASH TABLE SIZE
291      ;
292      000214 005767 000000G  HSH TBL: TST      .PNUMH      ; LEGAL HASH TABLE SIZE VALUE?
293      000220 001016      BNE      101$      ; BR IF NO
294      000222 016700      MOV      .PNUMB,R0      ; GET HASH TABLE SIZE
295      000226 012701 000000G  MOV      #HSHMN,R1      ; GET MINIMUM SIZE OF HASH TABLE
296      000232 020001      10$: CMP      R0,R1      ; IS THIS A LEGAL SIZE?
297      000234 001405      BEQ      20$      ; BR IF YES
298      000236 006301      ASL      R1      ; HASH TABLE SIZE MUST BE POWER OF TWO
299      000240 020127 001000      CMP      R1,#HSHMX      ; IS THIS A LEGAL SIZE?
300      000244 101004      BHI      101$      ; BR IF NO
301      000246 000771      BR      10$      ; CHECK FOR LEGAL VALUE
302      000250 010067 000000G  20$: MOV      R0,HSHSZ      ; SAVE HASH TABLE SIZE
303      000254      RETURN
304      ;
305      ; ERRORS
306      ;
307      000256      101$: MSG$R 04      ; ILLEGAL HASH TABLE SIZE
308      ;
309      ;
310      ; COUNTER TIMER VALUE
311      ;
312      000264 005767 000000G  CNTTIM: TST      .PNUMH      ; LEGAL COUNTER TIMER VALUE?
313      000270 001001      BNE      101$      ; BR IF NO
314      000272      RETURN
315      ;
316      ; ERRORS
317      ;
318      000274      101$: MSG$R 05      ; ILLEGAL COUNTER TIMER VALUE
319      ;

```

## MACRO CROSS REFERENCE

CREF 04.00

## MACRO NAME REFERENCES

|         |         |         |         |         |         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| ASL\$   | #5-60   | 20-500  |         |         |         |         |         |         |         |         |
| CALL    | 9-158   | 12-256  | 12-264  | 12-278  | 15-344  | 16-371  | 16-379  | 18-424  | 19-467  |         |
| DBGTP\$ | #10-173 | #10-185 | #10-204 | #10-208 | #10-212 | #10-219 | #11-237 |         |         |         |
| DTEDF\$ | #5-59   | 5-64    |         |         |         |         |         |         |         |         |
| EMSG\$R | #5-59   | 9-167   | 12-269  | 13-307  | 13-318  | 14-338  | 15-350  | 18-444  |         |         |
| ISTAT\$ | #5-60   | 10-173  |         |         |         |         |         |         |         |         |
| MTRANS  | #10-173 |         |         |         |         |         |         |         |         |         |
| NTLER\$ | #5-60   | 8-129   | 8-130   | 8-131   | 8-132   | 8-133   |         |         |         |         |
| PHBDF\$ | #5-60   | 5-65    |         |         |         |         |         |         |         |         |
| RESMAP  | #5-74   | 16-391  |         |         |         |         |         |         |         |         |
| RESRG   | #5-59   |         |         |         |         |         |         |         |         |         |
| RETC    | #5-59   | 16-390  |         |         |         |         |         |         |         |         |
| RETURN  | 9-162   | 12-257  | 12-265  | 12-287  | 13-303  | 13-314  | 15-346  | 16-393  | 17-414  | 18-426  |
|         | 18-433  | 18-440  | 19-471  | 20-508  |         |         |         |         |         |         |
| SAVMAP  | #5-70   | 16-372  |         |         |         |         |         |         |         |         |
| SAVRG   | #5-59   |         |         |         |         |         |         |         |         |         |
| SLTDF\$ | #5-59   | 5-66    |         |         |         |         |         |         |         |         |
| STATE\$ | #5-60   | 10-177  | #10-180 | #10-183 | #10-187 | #10-190 | #10-193 | #10-196 | #10-199 | #10-202 |
|         | #10-206 | #10-210 | #10-214 | #10-217 | 11-226  | #11-229 | #11-232 | #11-235 | #11-239 | #11-242 |
| SWSTK\$ | 16-371  |         |         |         |         |         |         |         |         |         |
| TRANS   | #5-60   | #10-178 | #10-181 | #10-184 | #10-185 | #10-188 | #10-191 | #10-194 | #10-197 | #10-200 |
|         | #10-203 | #10-204 | #10-207 | #10-208 | #10-211 | #10-212 | #10-215 | #10-218 | #10-219 | #11-227 |
|         | #11-230 | #11-233 | #11-236 | #11-237 | #11-240 |         |         |         |         |         |

TITLE CFGPVC - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

#### MODULE DESCRIPTION:

NLT - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

#### IDENT HISTORY:

- 1.00 27-OCT-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```

507 .SBTTL CHKPVC - CHECK FOR VALID PVC
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
001050 017700 000000G
001054 062700 000006
001060
001064
001070 011000
001072 001434
001074 010046
001076
001102 012600
001104 026760 000010' 000010
001112 001004
001114 026760 000000G 000012
001122 001413
001124 012702 000006
001130 012701 000036'
001134 010003
001136 062703 000002
001142 122123
001144 001351
001146 005302
001150 003374
001152
001164
001170

      CHKPVC - CHECK FOR VALID PVC
      INPUTS:
      NONE
      OUTPUTS:
      CARRY CLEAR - SPECIFIED PVC IS UNIQUE
      CARRY SET - SPECIFIED PVC IS NOT UNIQUE
      RO DESTROYED

CHKPVC: MOV @PSIPT,RO ; POINT TO PSI HOME BLOCK
      ADD #H$PVC,RO ; POINT TO PVC NAME BLOCK LISTHEAD

      SWSTK$ 50$ ; ENTER SYSTEM STATE
      SAVMAP ; SAVE CURRENT MAPPING
10$: MOV (RO),RO ; GET ADDRESS OF NEXT BLOCK IN LIST
      BEQ 40$ ; BR IF END OF LIST
      MOV RO,-(SP) ; SET UP UNMAPPED ADDRESS
      CALL $CEACK ; CONVERT TO MAPPED ADDRESS
      MOV (SP)+,RO ; RETRIEVE MAPPED ADDRESS
      CMP PVCLCN,C$LCN(RO) ; SAME CHANNEL NUMBER?
      BNE 15$ ; NO, OKAY
      CMP DTEDES,C$DTE(RO) ; SAME DTE AS WELL?
      BEQ 30$ ; YES- ILLEGAL
15$: MOV #PVNAMX,R2 ; GET NUMBER OF CHARACTERS IN CIRCUIT ID
      MOV #PVCNAM,R1 ; POINT TO SPECIFIED PVC CIRCUIT ID
      MOV RO,R3 ; COPY PVC BLOCK POINTER
      ADD #C$NAM,R3 ; POINT TO CIRCUIT ID FIELD
20$: CMPB (R1)+,(R3) ; IS THIS A UNIQUE CIRCUIT ID?
      BNE 10$ ; BR IF YES, SO FAR - CONTINUE SEARCH
      DEC R2 ; MORE CHARACTERS TO CHECK?
      RGT 20$ ; BR IF YES
30$: RETC ; ELSE ERROR - SET USER C-BIT
40$: RESMAP ; RESTORE PREVIOUS MAPPING
50$: RETURN

```

SYMBOL CROSS REFERENCE

REF 04.00

| SYMBOL | VALUE      | REFERENCES   |
|--------|------------|--|
| HSHSZ  | = ***** GX | 18-658 18-668  |
| HTIMR  | 000000 R   | #7-128 *13-409 21-818  |
| HBNPT  | 000022     | 16-567   |
| HSPTB  | 000020     | 14-485 16-568  |
| HSPVC  | 000006     | 13-454 15-523  |
| INIXCB | 001266 R   | 14-489 #17-602   |
| INVCIR | 001720 R   | 20-714 20-727 #20-732 20-739 20-741 20-753 20-755              |
| KSARS  | = ***** GX | 13-418 13-456 15-526 15-545 16-565 16-583 18-646 18-675 21-802 |
|        |            | 21-804 21-820  |
| LF.ACT | = 100000   | #5-70  |
| LF.BRO | = 000400   | #5-70  |
| LF.BWT | = 000007   | #5-70  |
| LF.ENA | = 002000   | #5-70  |
| LF.LPB | = 001000   | #5-70  |
| LF.MDC | = 000100   | #5-70  |
| LF.MFL | = 004000   | #5-70  |
| LF.MTP | = 000020   | #5-70  |
| LF.PAC | = 000200   | #5-70  |
| LF.RDY | = 040000   | #5-70  |
| LF.REA | = 010000   | #5-70  |
| LF.SER | = 000040   | #5-70  |
| LF.TIM | = 000010   | #5-70  |
| LF.UNL | = 020000   | #5-70  |
| LF.X2P | = 000000   | #5-70  |
| LLCTA  | = ***** GX | 21-782   |
| LNKEND | = ***** GX | 13-455   |
| LN.CLO | = 000000   | #5-70  |
| LN.DUM | = 000005   | #5-70  |
| LN.LOA | = 000004   | #5-70  |
| LN.LCO | = 000003   | #5-70  |
| LN.OAU | = 000001   | #5-70  |
| LN.OFF | = 000001   | #5-70  |
| LN.ON  | = 000000   | #5-70  |
| LN.OOP | = 000004   | #5-70  |
| LN.OPE | = 000001   | #5-70  |
| LN.REF | = 000002   | #5-70  |
| LN.SER | = 000002   | #5-70  |
| LN.STA | = 000017   | #5-70  |
| LN.SUB | = 000360   | #5-70  |
| LN.TRI | = 000006   | #5-70  |
| LTIMR  | 000002 R   | #7-129 *13-410 21-819  |
| LSCHTB | 000030     | 18-650   |
| L.COST | 000015     | #5-70  |
| L.CTL  | 000012     | #5-70 20-743   |
| L.CVA  | 177776     | #5-70  |
| L.DDM  | 000002     | #5-70  |
| L.DDS  | 000004     | #5-70  |
| L.DLC  | 000003     | #5-70  |
| L.DLM  | 000006     | #5-70  |
| L.DLS  | 000010     | #5-70  |
| L.FLG  | 000000     | #5-70  |
| L.KRBA | 000016     | #5-70  |

```
164 .SBTTL SUBEXPRESSIONS
165
166 ;; FLAG BIT DEFINITIONS PLUS TRAILING COMMA
167 ;;
168 STATES BITS
169 TRANS %LF,X2P%,BITS1,,FG.X2P,CFGFLG
170 TRANS $RAD50,BITS1
171 TRANS <'>,$EXIT
172 STATES BITS1
173 TRANS <'!>,BITS2
174 TRANS '+,BITS2
175 TRANS <'>,$EXIT
176 STATES BITS2
177 TRANS %LF,X2P%,BITS1,,FG.X2P,CFGFLG
178 TRANS $RAD50,BITS1
179
180 ;; CHECK FOR END OF SOURCE LINE
181 ;;
182 STATES END
183 TRANS <'>,$EXIT
184 TRANS $EOS,$EXIT
185
186 STATES
```

```

112 .SBTTL STA$DF STATE TABLE
113
114 000270 ISTAT$ STAST,STAKW
115
116 ; STATION TABLE DEFINITION (STA$DF)
117 ;
118 000270 STATES$ STADF
119 000270 TRANS$ %STA$DF%,1,SYNERR
120
121 000270 STATES$ ; STATION NUMBER
122 000270 TRANS$ $NUMBR,,ST.NUM
123
124 000270 STATES$
125 000270 TRANS$ <','>
126
127 000270 STATES$ ; PROCESS NAME
128 000270 TRANS$ $RAD50
129
130 000270 STATES$
131 000270 TRANS$ !END,$EXIT ; DON'T REQUIRE ANY MORE
132 000270 TRANS$ <','>
133
134 000270 STATES$
135 000270 TRANS$ !BITS
136
137 000270 STATES$ STADF2 ; STATION COST
138 000270 TRANS$ $NUMBR,,ST.CST
139
140 000270 STATES$
141 000270 TRANS$ !END,$EXIT ; DON'T REQUIRE APR
142 000270 TRANS$ <','>
143
144 000270 STATES$ ; ACTIVE RATIO
145 000270 TRANS$ $NUMBR,,ST.APR
146
147 000270 STATES$
148 000270 TRANS$ !END,$EXIT
149 000270 TRANS$ <','>
150
151 000270 STATES$
152 000270 TRANS$ $NUMBR,,ST.HTM ; HELLO TIMER
153
154 ;
155 ; CHECK FOR END OF LINE
156 ;
157 000270 STATES$ END
158 000270 TRANS$ <','>,$EXIT
159 000270 TRANS$ $EOS,$EXIT
160
161 ; FLAG BIT DEFINITIONS PLUS TRAILING COMMA
162 ;
163 000270 STATES$ BITS
164 000270 TRANS$ $RAD50,BITS1
165 000270 TRANS$ <','>,$EXIT
166 000270 STATES$ BITS1
167 000270 TRANS$ <','>,$BITS2
168 000270 TRANS$ '+,BITS2

```

```

105          ;
106          ; CONSTANT DEFINITIONS
107          ;
108          000007          MAXWND = 7          ; MAXIMUM WINDOW SIZE
109          .SBTTL LOCAL DATA
110
111          ;
112          ; LOCAL DATA
113          ;
114          000000          .PSECT DATA,D
115          000000          NEXT: .BLKW 1          ; ADDRESS TO STORE NEXT DTE DIGIT
116          000002          HTIMR: .BLKW 1          ; LOCAL COPY OF HELLO TIMER
117          000004          LTIMR: .BLKW 1          ; LOCAL COPY OF LISTEN TIMER
118          000006          PKTSVC: .BLKW 1          ; PACKET SIZE
119          000010          FLGSVC: .BLKW 1          ; FLAGS WORD
120          000012          RTYSVC: .BLKW 1          ; RETRY TIMER
121          000014          STASVC: .BLKW 1          ; STATION NUMBER
122          000016          SLTA: .BLKW 1          ; LOCAL COPY OF $SLTA
123          000020          OWNSVC: .BLKB 1          ; OWNER
124          000021          CTLSVC: .BLKB 1          ; CONTROLLER NUMBER
125          000022          WND SVC: .BLKB 1          ; WINDOW SIZE
126          000023          RCL SVC: .BLKB 1          ; RECALL TIMER
127          000024          COUNT: .BLKB 1          ; NUMBER OF DIGITS IN DTE ADDRESS
128          000017          DTELEN = 15.
129          000025          DTESVC: .BLKB DTELEN          ; DTE ADDRESS
130          000044'          DTEEND = .
131          000044          DTEPCK: .BLKB <DTELEN+1>/2          ; DTE ADDRESS (BCD)
132          000054          CVTBUF: .BLKB 1          ; CONVERSION BUFFER
133          .EVEN

```



585  
 586  
 587  
 588  
 589  
 590  
 591  
 592  
 593  
 594  
 595  
 596  
 597  
 598  
 599  
 600  
 601 001440 010501  
 602 001442 116702 000024'  
 603 001446 001423  
 604 001450 105021  
 605 001452 005300  
 606 001454 003375  
 607 001456 005001  
 608 001460 005305  
 609 001462 112400  
 610 001464 005701  
 611 001466 001007  
 612 001470  
 613 001500 005201  
 614 001502 005205  
 615 001504 000401  
 616 001506 005301  
 617 001510 150015  
 618 001512 005302  
 619 001514 003362  
 620 001516

.SBTTL PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT

PCKBCD - PACK A STRING OF DIGITS IN BCD FORMAT

INPUTS:

R0 - NUMBER OF BYTES IN PACKED ADDRESS  
 R4 - ADDRESS OF BUFFER CONTAINING ASCII DIGITS TO PACK (0=END OF BUFFER)  
 R5 - ADDRESS OF BUFFER TO STORE PACKED DATA  
 COUNT - NUMBER OF DIGITS SPECIFIED

OUTPUTS:

BUFFER SPECIFIED IN R5 CONTAINS DATA IN BCD FORMAT

```

PCKBCD: MOV     R5,R1      ; GET ADDRESS OF PACKED DATA
        MOV     COUNT,R2  ; GET NUMBER OF DIGITS IN STRING
        BEQ     40$       ; BR IF NOTHING TO DO
        CLR     (R1)+     ; INITIALIZE PACKED ADDRESS
        DEC     R0        ; MORE TO INITIALIZE?
        DEC     R1        ; BR IF YES
        CLR     R1        ; POSITION INDICATOR (0=HIGH FOUR BITS)
        DEC     R5        ; UPDATE ADDRESS FOR FIRST TIME
        MOV     (R4)+,R0  ; GET DIGIT TO PACK
        TST     R1        ; STORE IN HIGH FOUR BITS?
        BNE     20$       ; BR IF NO
        ASL     4,R0      ; MOVE TO HIGH FOUR BITS
        INC     R1        ; INDICATE STORE IN LOW FOUR BITS NEXT
        INC     R5        ; POINT TO NEXT BYTE
        BR      30$       ; BR TO STORE DECIMAL
        DEC     R1        ; INDICATE STORE IN HIGH FOUR BITS NEXT
        BIS     R0,(R5)   ; PACK THE NUMBER INTO THE BUFFER
        DEC     R2        ; MORE TO CONVERT?
        BGT     10$       ; BR IF YES
        RETURN
20$:    DEC     R1
30$:    BIS     R0,(R5)
40$:    RETURN
  
```

```
53      .SBTTL  MACRO DEFINITIONS
54      ;
55      ; LIBRARY MACROS
56      ;
57      .MCALL  EMSG$, ISTAT$, NTLER$, STATES$, TRANS$, SLTDF$, SAVRG, RESRG, CHADF$
58
59 000000      CHADF$                ; DEFINE CHARACTERISTICS WORD OFFSETS
60 000000      SLTDF$                ; DEFINE SLT OFFSETS
```

CFGUNT - CONFIG FILE SCAN ACTION MACRO V05.03b Monday 15-Jul-85 18:56 <sup>N 15</sup> Page 14-2  
Symbol table

Size of work file: 14981 Words ( 59 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:55.27

DB2:CFGUNT.153,[132,134]CFGUNT/CR/-SP=DB2:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFGUNT

B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z  
[  
,  
;  
:  
@  
#  
\$  
%  
&  
'  
(  
)  
\*  
+  
=  
>  
?  
/

CFGDTE

CFGPVC

CF GUNT

CF GX2P

```

132          .SBTTL X2P$DF STATE TABLE
133
134          : TPARS STATE TABLES
135          :
136 000216          ISTAT$ X2PST,X2PKW
137          :
138          : X2P$DF
139          :
140 000216          STATES$ X2PDF
141 000216          TRANS$ %X2P$DF%,.,.,1,SYNERR
142
143 000216          STATES$
144 000216          TRANS$ $.AD50,,X2.DEV          ; DEVICE NAME
145
146 000216          STATES$
147 000216          TRANS$ <'>->
148
149 000216          STATES$
150 000216          TRANS$ $NUMBR,,X2.CTL          ; CONTROLLER NUMBER
151
152 000216          STATES$
153 000216          TRANS$ <','>X2PRM,CHKDEV
154 000216          TRANS$ <'>->
155
156 000216          STATES$
157 000216          TRANS$ $NUMBR,,X2.UNT          ; UNIT NUMBER
158
159 000216          STATES$
160 000216          TRANS$ <'>
161
162 000216          STATES$ X2PRM
163 000216          TRANS$ $NUMBR,,X2.'WS          ; MAXIMUM WINDOW SIZE
164
165 000216          STATES$
166 000216          TRANS$ <'>
167
168 000216          STATES$
169 000216          TRANS$ $NUMBR,,X2.MBS          ; MAXIMUM BLOCK SIZE
170
171 000216          STATES$
172 000216          TRANS$ <'>
173
174 000216          STATES$
175 000216          TRANS$ $NUMBR,,X2.MRT          ; MAXIMUM RETRANSMIT COUNT
176
177 000216          STATES$
178 000216          TRANS$ <'>
179
180 000216          STATES$
181 000216          TRANS$ $NUMBR,,X2.RTR          ; RETRANSMIT TIMER
182
183 000216          STATES$
184 000216          TRANS$ <'>
185
186 000216          STATES$
187 000216          TRANS$ $NUMBR,,X2.HLD          ; HOLDBACK TIMER
188

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE U<sup>S</sup>E OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

1.00 02-JUN-81  
DECNET-11M/S V3.1  
DECNET-11M-PLUS V1.1

3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1

4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0

```
5.00 22-JUL-85
      DECnet-11M/S V4.2
      DECnet-11M-Plus V3.0
      DECnet-Micro/RX V1.0
```

\*\*FILE\*\*ID\*\*CF2AC1

|          |          |            |            |          |        |      |
|----------|----------|------------|------------|----------|--------|------|
| CCCCCCCC | FFFFFFFF | 222222     | AAAAAA     | CCCCCCCC | 11     |      |
| CCCCCCCC | FFFFFFFF | 222222     | AAAAAA     | CCCCCCCC | 11     |      |
| CC       | FF       | 22         | AA         | CC       | 11     |      |
| CC       | FF       | 22         | AA         | CC       | 11     |      |
| CC       | FF       | 22         | AA         | CC       | 11     |      |
| CC       | FF       | 22         | AA         | CC       | 11     |      |
| CC       | FFFFFFFF | 22         | AA         | CC       | 11     |      |
| CC       | FFFFFFFF | 22         | AA         | CC       | 11     |      |
| CC       | FF       | 22         | AAAAAAAAAA | CC       | 11     |      |
| CC       | FF       | 22         | AAAAAAAAAA | CC       | 11     |      |
| CC       | FF       | 22         | AA         | CC       | 11     |      |
| CC       | FF       | 22         | AA         | CC       | 11     |      |
| CCCCCCCC | FF       | 2222222222 | AA         | CCCCCCCC | 111111 | .... |
| CCCCCCCC | FF       | 2222222222 | AA         | CCCCCCCC | 111111 | .... |

|            |          |            |
|------------|----------|------------|
| LL         | SSSSSSSS | TTTTTTTTTT |
| LL         | SSSSSSSS | TTTTTTTTTT |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SSSSSS   | TT         |
| LL         | SSSSSS   | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LLLLLLLLLL | SSSSSSSS | TT         |
| LLLLLLLLLL | SSSSSSSS | TT         |

CF2A  
Symb

\$\$\$C  
 \$\$\$C  
 \$\$\$P  
 \$\$\$T  
 CERR  
 CFLI  
 \$\$\$C  
 \$\$\$C  
 \$\$\$R  
 DECF  
 DEV  
 DEVE  
 D\$AM  
 D\$AM  
 D\$AM  
 D\$BR  
 D\$BR  
 D\$DE  
 D\$DE  
 D\$EM  
 D\$EM  
 D\$H  
 D\$H  
 D\$IN  
 D\$IN  
 D\$IF  
 D\$IF  
 D\$LN  
 D\$LN  
 D\$LN

. AE  
 DATA  
 Error

\*\*\*

Work  
 Work  
 Size  
 Size  
 Oper

Elap  
 SY:

|                  |                    |                   |                  |                     |
|------------------|--------------------|-------------------|------------------|---------------------|
| ASSCHK= 000000   | D\$ST 000047       | GETSTR 000000RG   | N\$SBUF= 000001  | STRSIZ= ***** GX    |
| ASSCPS= 000000   | D\$MAXC 000064     | GRPUIC 000102RG   | N\$SLDV= 000001  | S\$WARG= 000000     |
| ASSPRI= 000000   | D\$MAXH 000066     | G\$STPP= 000000   | N\$MCP= 000001   | S\$VSZ= 007600      |
| ASSTRP= 000000   | D\$MAXV 000070     | G\$STSS= 000000   | N\$SML= 000001   | S.BRA 000740RG      |
| CERR = ***** GX  | D\$MLL 000040      | G\$STTK= 000000   | N\$SMOV= 000010  | S.CTIM 000714RG     |
| CFLIN = ***** GX | D\$MNOD 000041     | G\$SWRD= 000000   | N\$SNCT= 000001  | T\$KMG= 000000      |
| C\$CKCP= 000000  | D\$NA 000062       | I\$SAR= 000000    | N\$SPEM= 000001  | T\$MIN= 000000      |
| C\$SORE= 000400  | D\$NBEA 000056     | I\$SRDN= 000000   | P\$P45= 000000   | UICBUF= ***** GX    |
| C\$SRSH= 177564  | D\$NBRA 000054     | K\$AR5 = ***** GX | P\$SWRD= 000000  | USRUIC 000162RG     |
| DECP = ***** GX  | D\$NEND= 000054    | K\$CNT= 177546    | QUICK 000766R    | USRU11 000154RG     |
| DEV 000030RG     | D\$NLN 000030      | K\$CSR= 177546    | Q\$OPT= 000010   | V\$CTR= 001000      |
| DEVBUF= ***** GX | D\$NN 000060       | K\$LDC= 000000    | RADDR 000006R    | X\$DBT= 000000      |
| D\$AMXC 000072   | D\$OUTT 000043     | K\$TPS= 000074    | RF.LOO= 100000   | X\$EACX= ***** GX   |
| D\$AMXH 000074   | D\$RETF 000050     | LD\$LP = 000000   | RTSPC = ***** GX | \$CTIM = ***** GX   |
| D\$ANN 000000    | D\$RNN 000002      | L\$ASG= 000000    | R\$DER= 000000   | \$ERRYN 000222R 002 |
| D\$BRPR 000102   | D\$RTMR 000076     | L\$DRV= 000000    | R\$K11= 000001   | \$ERRYN 000164R 002 |
| D\$BRTM 000100   | D\$SEG 000036      | L\$P11= 000001    | R\$SND= 000000   | \$ERRYX 000132R 002 |
| D\$DELF 000045   | D\$SER 000032      | L\$11R= 000000    | R\$11M= 000000   | \$ERRYZ 000050R 002 |
| D\$DELEW 000046  | D\$SRL 000052      | M\$CRB= 000124    | R.ADD 000010     | \$ERRYZ 000012R 002 |
| D\$END = 000104  | D\$BUG= 177514     | M\$CRX= 000000    | R.ARE 000366RG   | \$HEADR= ***** GX   |
| D\$ENB 000034    | D\$ISK= 000000     | M\$FCS= 000000    | R.FLAG 000012    | \$LINKX= ***** GX   |
| D\$HIOR 000024   | D\$11= 000001      | M\$MGME= 000000   | R.LEN 000014     | \$LSTAD= ***** GX   |
| D\$HOST 000022   | D\$SYNC= 000000    | M\$NET= 000000    | R.LNK 000000     | \$LSTRM= ***** GX   |
| D\$INAC 000044   | D\$YNM= 000000     | M\$OVR= 000000    | R.NAM 000002     | \$XALOC= ***** GX   |
| D\$INCT 000042   | E\$XPR= 000000     | NAME 000000R 002  | R.NAME 000312RG  | .PNUMB= ***** GX    |
| D\$IPL 000051    | FMT8 = ***** GX    | NAMED 000010R 002 | R.NUM 000444RG   | .PNUMH= ***** GX    |
| D\$LID 000020    | FM.8 = 000000      | NBRA = ***** GX   | STRADD= ***** GX | .PSTCN= ***** GX    |
| D\$LNAM 000006   | FM\$BUFF= ***** GX | N\$ACC= 000001    | STRPT = ***** GX | .PSTPT= ***** GX    |
| D\$LNUM 000014   | F\$VL= 000001      |                   |                  |                     |

. ABS. 000104 000 (RW,I,GBL,ABS,OVR)  
001030 001 (RW,I,LCL,REL,CON)  
DATA 000266 002 (RW,D,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 12032 Words ( 47 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:14.38

SY:CF2AC1.V2,[132,134]CF2AC1/CR/-SP=SY:[1,1]RSXCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXCM/PA:1,[132,10]CF2AC1



```

Symbol table
A$$CHK= 000000      G$$TSS= 000000      M$$NET= 000000      O.LF  000012      T$$KMG= 000000
A$$CPS= 000000      G$$TTK= 000000      M$$OVR= 000000      O.LV  000000      T$$MIN= 000000
A$$PRI= 000000      G$$WRD= 000000      N$$ACC= 000001      O.MXC 000004      V$$CTR= 001000
A$$TRP= 000000      I$$RAR= 000000      N$$BUF= 000001      O.NAM 000006      X$$DBT= 000000
CERR = ***** GX  I$$RDN= 000000      N$$LDV= 100001      O.NAME 000046RG  $CAT5 = ***** GX
CFLIN = ***** GX K$AR5 = ***** GX  N$$MCP= 100001      O.NUM 000000RG  $CEACX= ***** GX
C$CKP= 000000      K$CNT= 177546      N$$MLL= 000001      O.TYP 000002      $ERRXW 000010R  002
C$ORE= 000400      K$CSR= 177546      N$$MOV= 000010      O.VFY 000120RG  $ERRXX 000046R  002
C$RSH= 177564      K$LDL= 000000      N$$NCT= 000001      P$SP45= 000000      $ERRXY 000110R  002
D$BUG= 177514      K$TPS= 000074      N$$PEM= 000001      P$WRD= 000000      $ERRXZ 000154R  002
D$ISK= 000000      LD$LP= 000000      OBJHD = ***** GX  Q$OPT= 000010      $ERRYY 000224R  002
D$LLI= 000001      L$ASS= 000000      OF.PRO= 000040      RTSPC = ***** GX  $ERRYZ 000306R  002
D$YNC= 000000      L$DRV= 000000      OF.RLU= 000100      R$DER= 000000      $HEADR= ***** GX
D$YNM= 000000      L$P11= 000001      OF.SMC= 000200      R$SK11= 000001      $LINKX= ***** GX
E$XPR= 000000      L$11R= 000000      ONAME 000002R  002  R$SND= 000000      $XALDC= ***** GX
FLAG 000001R  002 MXCP 000006R  002 O.CPY 000202RG  R$11M= 000000      .PNUMB= ***** GX
FMT8 = ***** GX M$CRB= 000124      O.END 000224RG  S$WRG= 000000      .PNUMH= ***** GX
FM.8 = 000000      M$CRX= 000000      O.FLAG 000146RG  S$YSZ= 00 600      .PSTCN= ***** GX
F$LVL= 000001      M$FCS= 000000      O.FLG 000003      TYPE 000000R  002 .PSTPT= ***** GX
G$TPP= 000000      M$MGE= 000000

```

```

. ABS. 000012 000 (RW,I,GBL,ABS,OVR)
      000420 001 (RW,I,LCL,REL,CON)
DATA 000350 002 (RW,D,LCL,REL,CON)
Errors detected: 0

```

### \*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 10636 Words ( 42 Pages)
Size of core pool: 14440 Words ( 55 Pages)
Operating system: RSX-11M/PLUS

```

Elapsed time: 00:00:11.46

SY:CF2AC2.V2,C132,134]CF2AC2/CR/-SP=SY:[1,1]RSXMCM.SML/ML,C130,110]NETLIB/ML,C130,10]RSXMCM/PA:1,C132,10]CF2AC2

TPARS ACTION ROUTINES FOR ROU\$DF

```

296
297
298 000572 005767 000000G      R.BTM:: TST      .NN      ; ENDNODE ?
299 000576 001004      BNE      10$      ; IF NOT - BRANCH
300 000600 005767 000000G      TST      .PNUMB     ; CHECK FOR LEGAL VALUE
301 000604 001405      BEQ      20$      ; IF OK - BRANCH
302 000606 000410      BR       101$     ; IF NOT - ERROR
303 000610 022767 000377 000000G 10$: CMP      #255,,.PNUMB ; CHECK LIMIT
304 000616 002404      BLT      101$     ; IF ILLEGAL BRANCH
305
306 000620 016767 000000G 000000G 20$: MOV      .PNUMB,.ROUTB ; STORE BROADCAST TIMER VALUE
307 000626      RETURN
308
309 000630      101$:  EMSG$R  YI      ; ILLEGAL BROADCAST TIMER VALUE
310
311
312

```

.SBTTL ACTION ROUTINES TO SET UP EVENT FILTER CONTROL BLOCK

```

;+
; E.CON - EVENT IS DESTINED FOR CONSOLE
; E.FIL - EVENT IS DESTINED FOR FILE
; E.MON - EVENT IS DESTINED FOR A LOGGING MONITOR
;-

```

.ENABL LSB

```

161
162
163
164
165
166
167
168
169
170
171
172
173 000130 052767 000001 000002' E.CON:: BIS #FF.CON,FLAG ; INDICATE EVENT DESTINED FOR CONSOLE
174 000136 000407 BR 5$
175
176 000140 052767 000002 000002' E.FIL:: BIS #FF.FIL,FLAG ; INDICATE EVENT DESTINED FOR FILE
177 000146 000403 BR 5$
178
179 000150 052767 000004 000002' E.MON:: BIS #FF.MON,FLAG ; INDICATE EVENT DESTINED FOR CONSOLE
180
181 000156 012701 000016 5$: MOV #F.LEN,R1 ; GET LENGTH OF A FILTER CONTROL BLOCK
182
183 000162 SWSTK$ 60$ ;* ENTER KERNEL MODE
184 000166 CALL $XALOC ;* ATTEMPT TO ALLOCATE BLOCK
185 000172 103006 BCC 10$ ;* BR IF SUCCESSFUL
186 000174 RETC R0 ;* ELSE SET USER C-BIT
187 000206 000461 BR 50$ ;* AND EXIT
188 000210 10$: SAVMAP ;* SAVE CURRENT MAPPING
189 000214 010002 MOV R0,R2 ;* SAVE BLOCK ADDRESS
190 000216 010046 MOV R0,-(SP) ;* SET BLOCK ADDRESS
191 000220 CALL $CEACK ;* GET ACCESS TO BLOCK
192 000224 012600 MOV (SP)+,R0 ;* GET MAPPED ADDRESS OF BLOCK
193 000226 01676C 000000' 000002 MOV CLASS,F.CLS(R0) ;* STORE THE EVENT CLASS
194 000234 052760 000016 000002 BIS #F.LEN,F.CLS(R0) ;* STORE LENGTH OF BLOCK
195 000242 016760 000004' 000004 MOV EVENT,F.EVT(R0) ;* STORE THE EVENT TYPE MASKS
196 000250 016760 000006' 000006 MOV EVENT+2,F.EVT+2(R0) ;* ...
197 000256 016760 000010' 000010 MOV EVENT+4,F.EVT+4(R0) ;* ...
198 000264 016760 000012' 000012 MOV EVENT+6,F.EVT+6(R0) ;* ...
199 000272 016760 000002' 000014 MOV FLAG,F.FLG(R0) ;* STORE FLAGS
200 000300 016700 000000G MOV FILHD,R0 ;* GET ADDRESS OF BEGINNING OF LIST
201 000304 010003 MOV R0,R3 ;* SAVE ADDRESS OF PREVIOUS BLOCK
202 000306 010301 20$: MOV R3,R1 ;* SAVE UNMAPPED ADDRESS OF PREVIOUS BLOCK
203 000310 011000 MOV (R0),R0 ;* GET NEXT BLOCK IN LIST
204 000312 001411 BEQ 40$ ;* IF EQ, INSERT AT END OF LIST
205 000314 010003 MOV R0,R3 ;* SAVE ADDRESS OF PREVIOUS BLOCK
206 000316 010046 MOV R0,-(SP) ;* SET ADDRESS OF BLOCK
207 000320 CALL $CEACK ;* GET ACCESS TO BLOCK
208 000324 012600 MOV (SP)+,R0 ;* GET MAPPED ADDRESS OF BLOCK
209 000326 026760 000000' 000002 CMP CLASS,F.CLS(R0) ;* LINK FILTER CONTROL BLOCK HERE ?
210 000334 101364 BHI 20$ ;* BR IF NO
211 000336 010246 40$: MOV R2,-(SP) ;* SET BLOCK ADDRESS
212 000340 010146 MOV R1,-(SP) ;* SET ADDRESS OF PREVIOUS BLOCK IN LIST
213 000342 CALL $LINKX ;* LINK BLOCK INTO LIST
214 000346 RESMAP ;* RESTORE PREVIOUS MAPPING
215 000352 RETURN ;* RETURN TO USER MODE AND TO CALLER
216
217 000354 103401 60$: BCS 101$ ;* BR IF UNABLE TO ALLOCATE BLOCK

```

★★FII

|  |     |
|--|-----|
|  | CCC |
|  | CCC |

CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC

|    |    |
|----|----|
| CC | CC |
| CC | CC |

[illegible]

|          |              |            |              |          |          |    |
|----------|--------------|------------|--------------|----------|----------|----|
| CCCCCCCC | FFF, FFFFFFF | 222222     | AAAAAA       | CCCCCCCC | 666666   |    |
| CCCCCCCC | FFFFFFFFFFFF | 222222     | AAAAAA       | CCCCCCCC | 666666   |    |
| CC       | FF           | 22         | AA           | CC       | 66       |    |
| CC       | FF           | 22         | AA           | CC       | 66       |    |
| CC       | FF           | 22         | AA           | CC       | 66       |    |
| CC       | FF           | 22         | AA           | CC       | 66       |    |
| CC       | FFFFFFFFFFFF | 22         | AA           | CC       | 66666666 |    |
| CC       | FFFFFFFFFFFF | 22         | AA           | CC       | 66666666 |    |
| CC       | FF           | 22         | AAAAAAAAAAAA | CC       | 66       |    |
| CC       | FF           | 22         | AAAAAAAAAAAA | CC       | 66       | 66 |
| CC       | FF           | 22         | AAAA         | CC       | 66       | 66 |
| CC       | FF           | 22         | AA           | CC       | 66       | 66 |
| CC       | FF           | 22         | AA           | CC       | 66       | 66 |
| CCCCCCCC | FF           | 2222222222 | AA           | CCCCCCCC | 666666   |    |
| CCCCCCCC | FF           | 2222222222 | AA           | CCCCCCCC | 666666   |    |

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSSSS  TT
LL          SSSSSSSS  TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT

```

```

Symbol table
A$$CHK= 000000
A$$CPS= 000000
A$$PRI= 000000
A$$TRP= 000000
CERR = ***** GX
CFLIN = ***** GX
CTL 000003R
C$$CKP= 000000
C$$ORE= 000400
C$$RSH= 177564
DECP = ***** GX
D$AMXC 000072
D$AMXH 000074
D$ANN 000000
D$BPR 000102
D$BRTM 000100
D$DEL 000045
D$DELW 000046
D$END = 000104
D$FNB 000034
D$HIOP 000024
D$HOST 000022
D$INACT 000044
D$INACT 000042
D$IPL 000051
D$LID 000020
D$LNAM 000006
D$LNUM 000014
D$LST 000047
D$MAX 000064
D$MAXH 000066
D$MAXV 000070
D$MLL 000040
D$MNOD 000041
D$NA 000062
D$NBEA 000056
D$NBRA 000034
D$NEND= 000054
D$NLN 000030
D$NN 000060
D$OUT 000043

D$RETF 000050
D$RNN 000002
D$RTM 000076
D$SEG 000036
D$SER 000032
D$SRL 000052
D$BUG= 177514
D$ISK= 000000
D$L11= 000001
D$SYIC= 000000
D$YNM= 000000
F$XPR= 000000
FMT8 = ***** GX
FM.8 = 000000
F$LVL= 000001
G$TPP= 000000
G$TSS= 000000
G$TTK= 000000
G$WRD= 000000
H$TARE 000454RG
H$TNM 000526RG
I$RAR= 000000
I$RDN= 000000
K$ARS = ***** GX
K$CNT= 177546
K$CSR= 177546
K$LDC= 000000
K$TPS= 000074
LD$LP = 000000
LF.ACT= 100000
LF.BRO= 000400
LF.BWT= 000007
LF.ENA= 002000
LF.LPB= 001000
LF.MDC= 000100
LF.MFL= 004000
LF.MTP= 000020
LF.PAC= 000200
LF.RDY= 040000
LF.REA= 010000
LF.SER= 000040

LF.TIM= 000010
LF.UNL= 020000
LF.X2P= 000000
LGSTT = ***** GX
LN.CLO= 000000
LN.DUM= 000005
LN.LOA= 000004
LN.LOO= 000003
LN.OAU= 000003
LN.OFF= 000001
LN.ON = 000000
LN.OOP= 000004
LN.OPE= 000001
LN.REF= 000002
LN.SER= 000002
LN.STA= 000017
LN.SUB= 000360
LN.TRI= 000006
L$ASG= 000000
L$DRV= 000000
L$P11= 000001
L$1R= 000000
L.COST 000015
L.CTL 000012
L.CVA 177776
L.DDM 000002
L.DDS 000004
L.DLC 000003
L.DLM 000006
L.DLS 000010
L.FLG 000000
L.KRBA 000016
L.LEN = 000022
L.LGST 001232RG
L.MPF 000022
L.NMST 000020
L.NSTA 000014
L.OWNR 000021
L.UNT 000013
MRKFL = ***** GX
M$CRB= 000124

M$CRX= 000000
M$FCS= 000000
M$MGE= 000000
M$NET= 000000
M$OVR= 000000
NAME 000000R
NODARE 000356RG
NODID 000024RG
NODNM 000000RG
NODNUM 000430RG
N$ACC= 000001
N$BUF= 000001
N$LDV= 000001
N$MCP= 000001
N$MLL= 000001
N$MOV= 000010
N$NCT= 000001
N$PEM= 000001
P$VNM = ***** GX
PRMFL 000644RG
PRONM 000560RG
P$P45= 000000
P$WRD= 000000
Q$OPT= 000010
RTSPC = ***** GX
R$DER= 000000
R$K11= 000001
R$SND= 000000
R$TIM= 000000
SF.ACT= 000000
SF.ENA= 000100
SF.LPB= 000004
SF.MFL= 000040
SF.PAC= 000020
SF.REA= 000010
SF.SER= 000001
SF.SVC= 000002
SF.UNL= 000040
SLTMA = ***** GX
SLTNM = ***** GX

STAN = ***** GX
STRAD 000006R
STRCH 000154RG
STRCT 000012R
STRMX 000010R
STRPT 000014R
STRXX 000016R
S$WRG= 000000
S$YSZ= 007600
S.COST 000001
S.CTL 000700RG
S.FLG 000000
S.LEN 000004
S.NAME 000660RG
S.NMST 000002
S.OWNR 000003
S.UNT 000710RG
T$KMG= 000000
T$MIN= 000000
UNT 000004R
V$CTR= 001000
X$DBT= 000000
X.FLG 001124RG
Z.F.MFL= ***** GX
Z.FLG = ***** GX
Z.NAM = ***** GX
$CAT5 = ***** GX
$CEACX= ***** GX
$ERR1 000234R
$ERRN 000020R
$ERRP 000056R
$ERRXT 000112R
$ERRXU 000142R
$ERRYM 000176R
$PDVA = ***** GX
$SLTA = ***** GX
$XALOC= ***** GX
.CBIAS= ***** GX
.PNUMB= ***** GX
.PSTPT= ***** GX

```

```

. ABS. 177776 000 (RW,I,BGL,ABS,OVR)
      001270 001 (RW,I,LCL,REL,CON)
DATA 000272 002 (RW,D,LCL,REL,CON)
Errors detected: 0

```

\*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 13564 Words ( 53 Pages)
Size of core pool: 14440 Words ( 55 Pages)
Operating system: RSX-11M/PLUS

```

```

Elapsed time: 00:00:16.31
SV:CF2AC6.V2,[132,134]CF2AC6/CR/-SP=SV:[1,1]RSX-MCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSX-MCM/PA:1,[132,10]CF2AC6

```

```

302      MOV      .CRISK,R0      ; GET ADDRESS OF MY ATL NODE
303      MOV      A.HA(R0),R0    ; GET ADDRESS OF TASK HEADER
304      CALL     $MAPX          ; MAP TO TASK HEADER
305      MOV      .RGID,R3       ; GET REGION ID
306      DEC      R3             ; START COUNTING AT ZERO
307      ASH      #2,R3          ; FORM OFFSET
308      ADD      H.ADB(R2),R3    ; COMPUTE OFFSET INTO HEADER
309      BIC      #160000,R5      ; REMOVE APR BIAS
310      BIS      R2,R3          ; AND USE CORRECT APR
311      MOV      (R3),R4         ; GET ADDRESS OF GCD NODE
312      MOV      R4,.REG         ; SAVE ADDRESS OF GCD FOR LATER
313      INCB     G.AC(R4)        ; LOCK REGION IN MEMORY
314      INCB     G.IC(R4)
315      MOV      G.PD(R4),R3     ; GET TPD LISTHEAD FOR REGION
316      CALL     .MUF1          ; FIND ASSOCIATED MUL ENTRY
317      BCS      20$            ; IF CS; NOT FOUND
318      INCB     M.IO(R3)        ; LOCK TIMESHARING REC'ON IN MEMORY
319      CALL     @ (SP)+         ; CO-ROUTINE RETURN TO RESET MAPPING
320      MOV      G.BA(R4),.HOME  ; SAVE HOME BLOCK BIAS
321      MOV      R4,.BASEB       ; STORE POOL GCD ADDRESS IN HOME BLOCK
322      MOV      (SP)+,R4        ; RESTORE REGISTERS
323      MOV      (SP)+,R3        ; ...
324      RETURN
325
326      101$:     MSG$R XJ        ; FAILED TO CREATE REGION
327      102$:     MSG$R XK        ; MAPPING FAILED
328      .ENDC
329      .ENDC

```

```

90
91          .SBTTL  SERSLN - DETERMINE SLN/STA VALUE
92
93          ;+
94          SERSLN - DETERMINE SLN/STA VALUE
95          ;
96          : INPUTS -
97          : CIRNAM - DEVICE NAME (ASCII STRING)
98          : CIRNUM - DEVICE UNIT/CONTROLLER NUMBER
99          : CIRMUL - DEVICE TRIB NUMBER
100
101          : OUTPUTS -
102          : SERSLT - CONTAINS SLN/STA VALUE
103          :-
104
105 000000 010346 000000S SERSLN::MOV R3,-(SP) ; SAVE REG
106 000002 012700 000000S MOV #CIRNAM,R0 ; GET DEVICE STRING
107 000006 CALL $CAT5 ; CONVERT TO RAD50
108 000012 CALL FPDV ; FIND DEVICE DDM PDV INDEX
109
110 000016 010100 MOV R1,R0 ; GET DDM PDV INDEX
111 000020 005001 CLR R1 ; GET CONTROLLER NUMBER
112
113 000022 116701 000000G MOVB CIRNUM,R1 ; GET UNIT NUMBER
114 000026 005002 CLR R2 ; GET UNIT NUMBER
115 000030 116702 000000G MOVB CIRUNT,R2 ; GET SLN
116 000034 CALL FSLT ; GET SLN
117
118 000040 010367 000000G MOV R3,SERSLT ; STORE SLN
119 000044 116767 000000G 000001G MOVB CIRMUL,SERSLT+1 ; STORE STATION NUMBER
120 000052 012603 MOV (SP)+,R3 ; RESTORE REG
121
122 000054 RETURN
  
```



CF2AC8 CREATED BY MACRO ON 29-JUN-85 AT 00:12

PAGE 4 B 13

SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE      | REFERENCES         |
|---------|------------|--------------------|
| ZF.TIM  | = 000200   | #5-61              |
| ZF.X3P  | = 000000   | #5-61              |
| ZS.ASN  | = 100000   | #5-61              |
| ZS.BSY  | = 140000   | #5-61              |
| Z.AVL   | 000014     | #5-61              |
| Z.DAT   | 000016     | #5-61              |
| Z.DSP   | 000000     | #5-61 5-61         |
| Z.FLG   | 000010     | #5-61              |
| Z.LEN   | = 000016   | #5-61              |
| Z.LLN   | 000006     | #5-61              |
| Z.MAP   | 000020     | #5-61              |
| Z.NAM   | 000004     | #5-61 10-354       |
| Z.PCB   | 000012     | #5-61              |
| Z.SCH   | 000007     | #5-61              |
| \$CAT5  | = ***** GX | 6-107              |
| \$CEACX | = ***** GX | 8-199              |
| \$ERRSS | 000000 R   | #5-85 8-238 8-183  |
| \$HEADR | = ***** GX | 8-191              |
| \$LINKX | = ***** GX | 8-244              |
| \$XALOC | = ***** GX | 8-188              |
| .PSTCN  | = ***** GX | 9-275 10-330       |
| .PSTPT  | = ***** GX | 8-157 9-276 10-329 |

CF2  
SER

```

322 .SBTTL SER$DF ACTION ROUTINES
323
324 SER$DF ACTION ROUTINES
325
326 INPUTS -
327 .PNUMB - NUMERICAL VALUE PARSED
328 .PSTPT - PARSED STRING ADDRESS
329 .PSTCN - STRING SIZE
330
331 OUTPUTS -
332 PARAMETERS COPIED INTO NTLCF2
333
334 000652 016767 000000G 000000G SERUNI: MOV .PNUMB,CIRNUM ; STORE UNIT NUMBER
335 000660 RETURN
336
337 000662 016767 000000G 000000G SERMUL: MOV .PNUMB,CIRMUL ; STORE TRID ADDRESS
338 000670 RETURN
339
340 000672 016767 000000G 000000G SERMUX: MOV .PNUMB,CIRUNT ; STORE UNIT NUMBER
341 000700 RETURN
342
343 000702 016700 000000G NOAR: MOV .PNUMB,R0 ; STORE NODE AREA
344 000706 012701 002000 MOV #1024.,R1 ; POSITION AREA ADDRESS
345 000712 070100 MUL R0,R1 ; ...
346 000714 010167 000000G MOV R1,SERNOD ; SAVE NODE AREA
347 000720 RETURN
348
349 000722 056767 000000G 000000G NOAD: BIS .PNUMB,SERNOD ; SET NODE ADDRESS
350 000730 RETURN
351
352 000732 016700 000000G HSAR: MOV .PNUMB,R0 ; SET HOST AREA
353 000736 012701 002000 MOV #1024.,R1 ; POSITION AREA ADDRESS
354 000742 070100 MUL R0,R1 ; ...
355 000744 010167 000000G MOV R1,SERHST ; SAVE HOST AREA
356 000750 RETURN
357
358 000752 056767 000000G 000000G HSAD: BIS .PNUMB,SERHST ; SET HOST ADDRESS
359 000760 RETURN
360
361 000762 016700 000000G SERDV: MOV .PSTPT,R0 ; GET STRING ADDRESS
362 000764 000764 CALL $CAT5 ; CONVERT TO RAD50
363 000772 010167 000000G MOV R1,SERDEV ; STORE DEVICE NAME
364 RETURN
365
366 001000 016767 000000G 000000G DADDR: MOV .PNUMB,SERDMP ; STORE DUMP ADDRESS
367 001006 016767 000000G 000002G MOV .PNUMH,SERDMP+2 ; ...
368 RETURN
369
370 001014
371
372 001016 016767 000000G 000000G DCT: MOV .PNUMB,SERDCT ; STORE DUMP COUNT
373 001024 RETURN
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
  
```

```

;+
; STORE NUMBER OF XPT CIRCUITS
;-
  
```

CF2AC9      CREATED BY    MACRO    ON 29-JUN-85 AT 00:13      PAGE 6      B 15

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |         |         |         |         |         |         |         |         |         |
|------------|------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| CALL       | 7-97       | 7-118   | 7-124   | 7-158   | 7-204   | 7-207   | 9-362   |         |         |         |
| DBGTP\$    | #10-398    | #10-417 | #10-420 | #10-421 | #10-422 | #10-425 | #10-434 | #10-437 | #10-440 | #10-443 |
|            | #10-446    | #10-449 | #10-452 | #11-468 | #11-469 |         |         |         |         |         |
| DHBD\$     | #5-54      | 6-62    |         |         |         |         |         |         |         |         |
| EMSG\$     | #5-54      | 7-103   | 7-172   | 7-252   |         |         |         |         |         |         |
| FNBD\$     | #5-54      | 6-63    |         |         |         |         |         |         |         |         |
| ISTAT\$    | #5-55      | 10-398  |         |         |         |         |         |         |         |         |
| MTRAN\$    | #10-398    |         |         |         |         |         |         |         |         |         |
| NTLR\$     | #5-54      |         |         |         |         |         |         |         |         |         |
| PDVDF\$    | #5-55      | 6-65    |         |         |         |         |         |         |         |         |
| RESMAP     | #5-54      |         |         |         |         |         |         |         |         |         |
| RETC       | #5-54      |         |         |         |         |         |         |         |         |         |
| RETURN     | 7-101      | 7-155   | 7-182   | 7-210   | 7-219   | 7-254   | 7-259   | 8-284   | 8-315   | 8-318   |
|            | 9-335      | 9-338   | 9-341   | 9-347   | 9-350   | 9-356   | 9-359   | 9-365   | 9-370   | 9-373   |
|            | 9-381      | 9-384   | 9-387   | 9-390   |         |         |         |         |         |         |
| SAVMAP     | #5-54      |         |         |         |         |         |         |         |         |         |
| SERDF\$    | #5-54      | 6-64    |         |         |         |         |         |         |         |         |
| SLIDF\$    | #5-55      | 6-66    |         |         |         |         |         |         |         |         |
| SOB        | 7-201      |         |         |         |         |         |         |         |         |         |
| STATE\$    | #5-55      | 10-401  | #10-403 | #10-405 | #10-407 | #10-409 | #10-411 | #10-413 | #10-415 | #10-418 |
|            | #10-423    | #10-426 | #10-428 | #10-430 | #10-432 | #10-435 | #10-438 | #10-441 | #10-444 | #10-447 |
|            | #10-450    | #10-453 | #11-457 | #11-460 | #11-463 | #11-466 | #11-471 | #11-474 | #11-477 | #12-487 |
|            | #12-489    | #12-491 | #12-493 | #12-495 |         |         |         |         |         |         |
|            | 7-124      |         |         |         |         |         |         |         |         |         |
| SWSTK\$    | #5-55      | #10-402 | #10-404 | #10-406 | #10-408 | #10-410 | #10-412 | #10-414 | #10-416 | #10-417 |
| TRANS      | #10-419    | #10-420 | #10-421 | #10-422 | #10-424 | #10-425 | #10-427 | #10-429 | #10-431 | #10-433 |
|            | #10-434    | #10-436 | #10-437 | #10-439 | #10-440 | #10-442 | #10-443 | #10-445 | #10-446 | #10-448 |
|            | #10-449    | #10-451 | #10-452 | #10-454 | #11-458 | #11-461 | #11-464 | #11-467 | #11-468 | #11-469 |
|            | #11-472    | #11-475 | #11-478 | #12-488 | #12-490 | #12-492 | #12-494 |         |         |         |

CF2DEC CREATED BY MACRO ON 29-JUN-85 AT 00:14

PAGE 3 B 16

MACRO CROSS REFERENCE

CREF 04.00

MACRO NAME REFERENCES

|         |        |        |        |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| CALL    | 6-68   |        |        |        |        |        |        |        |        |        |
| DBGTP\$ | #8-177 | #8-181 |        |        |        |        |        |        |        |        |
| DHBDP\$ | #5-50  | 5-57   |        |        |        |        |        |        |        |        |
| EMSGBR  | #5-50  | 6-74   | 7-91   | 7-100  | 7-108  | 7-116  | 7-124  | 7-132  | 7-140  | 7-148  |
|         | 7-157  | 7-166  |        |        |        |        |        |        |        |        |
| ISTAT\$ | #5-50  | 8-177  |        |        |        |        |        |        |        |        |
| MTRAN\$ | #8-177 |        |        |        |        |        |        |        |        |        |
| RETURN  | 6-72   | 7-89   | 7-98   | 7-106  | 7-114  | 7-122  | 7-130  | 7-138  | 7-146  | 7-155  |
|         | 7-164  | 7-170  |        |        |        |        |        |        |        |        |
| STAT\$  | #5-50  | 8-179  | #8-183 | #8-186 | #8-189 | #8-192 | #8-195 | #8-198 | #8-201 | #8-204 |
|         | #8-207 | #8-210 | #8-213 | #8-216 | #8-219 | #8-222 | #8-225 | #8-228 | #8-231 | #8-234 |
|         | #8-237 | #8-240 | #8-243 | #8-246 |        |        |        |        |        |        |
| TRANS   | #5-50  | #8-180 | #8-181 | #8-184 | #8-187 | #8-190 | #8-193 | #8-196 | #8-199 | #8-202 |
|         | #8-205 | #8-208 | #8-211 | #8-214 | #8-217 | #8-220 | #8-223 | #8-226 | #8-229 | #8-232 |
|         | #8-235 | #8-238 | #8-241 | #8-244 |        |        |        |        |        |        |

|     |        |         |              |
|-----|--------|---------|--------------|
| 189 | 000216 | STATE\$ |              |
| 190 | 000216 | TRANS   | <'>,\$EXIT   |
| 191 | 000216 | TRANS   | \$EOS,\$EXIT |
| 192 |        |         |              |
| 193 | 000216 | STATE\$ |              |

5  
5  
5  
5  
5  
6

```
55          .SBTTL  MACRO DEFINITIONS
56
57          ;
58          ; LIBRARY MACROS
59          ;
60          .MCALL  ISTAT$,STATE$,TRANS$,EMSG$R,NTLER$
```

CF2AC1 - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:09 <sup>C 3</sup>  
 Table of contents

|     |     |  |
|-----|-----|--|
| 5-  | 56  | MACRO DEFINITIONS                        |
| 6-  | 78  | LOCAL DATA                               |
| 7-  | 91  | ERROR MESSAGES                           |
| 8-  | 111 | FILEDF ACTION ROUTINES                   |
| 11- | 325 | COUNTER TIMER ACTION ROUTINE             |
| 12- | 346 | QUICK - FAST REMOTE NODE LIST PROCESSING |

CF2A  
 SYMB  
 SYMB  
 CERR  
 CFLI  
 DECP  
 DEV  
 DEVE  
 DSRN  
 FMT8  
 FM.8  
 FNBL  
 GETS  
 GRPL  
 ISSA  
 KSAF  
 NAME  
 NAME  
 NBR  
 NSS  
 QUIC  
 RADDD  
 RF.L  
 RTS  
 R\$SE  
 R\$E  
 R.AL  
 R.AA  
 R.FL  
 R.LL  
 R.LN  
 R.NA  
 R.NA  
 R.NU  
 STR  
 STR  
 STR  
 S\$SE  
 S.BI  
 S.C  
 UIC  
 USRI  
 USRI  
 \$CE  
 \$CT  
 \$ER  
 \$ER  
 \$ER  
 \$ER  
 \$HE  
 \$LI  
 \$LS

CF2AC1      CREATED BY    MACRO    ON 29-JUN-85 AT 00:09      PAGE 1      C 4

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|---|
| CERR    | = ***** GX  | 7-99      7-100      7-101      7-102      7-103  |
| CFLIN   | = ***** GX  | 7-103   |
| DECP    | = ***** GX  | 10-291  |
| DEV     | = 000030 RG | #8-142  |
| DEVBUF  | = ***** GX  | *8-144      8-145      *8-151   |
| DERRN   | = 000002    | 10-292  |
| FMT8    | = ***** GX  | 7-99      7-100      7-101      7-102      7-103  |
| FM.8    | = 000000    | #7-99      #7-100      #7-101      #7-102      #7-103   |
| FNBUFF  | = ***** GX  | 8-152   |
| GETSTR  | = 000000 RG | #8-122  |
| GRPUIC  | = 000102 RG | #9-171  |
| ISSAS   | = *****     | 10-259  |
| KSARS   | = ***** GX  | 10-276  |
| NAME    | = 000000 R  | #6-87      10-238      10-244      10-281      10-282      10-283                                   |
| NAMED   | = 000010 R  | #6-89      *9-199      9-202      9-213      *9-219   |
| NBRA    | = ***** GX  | *11-339   |
| NS\$VCT | = *****     | 10-274  |
| QUICK   | = 000766 R  | 10-287      #12-362   |
| RADDR   | = 000006 R  | #6-88      *10-260      *10-269      10-284      10-300      10-307      12-368      12-372         |
| RF.LOO  | = 100000    | #5-65   |
| RTSPC   | = ***** GX  | 7-99      7-100      7-101      7-102      7-103  |
| R\$EIS  | = *****     | 10-259  |
| R\$E11D | = *****     | 10-259  |
| R.ADD   | = 000010    | #5-65      *10-284      10-307  |
| R.ARE   | = 000366 RG | #10-256   |
| R.FLAG  | = 000012    | #5-65      *10-285  |
| R.LEN   | = 000014    | #5-65      10-270   |
| R.LNK   | = 000000    | #5-65   |
| R.NAM   | = 000002    | *10-281      *10-282      *10-283   |
| R.NAME  | = 000312 RG | #10-238   |
| R.NUM   | = 000444 RG | #10-267   |
| STRADD  | = ***** GX  | 8-142      9-172      9-201   |
| STRPT   | = ***** GX  | 8-123      *8-124      *8-152      *9-183      *9-228   |
| STRSIZ  | = ***** GX  | *8-125      8-143      *8-153      9-171      *9-184      9-200      *9-229                         |
| S\$BAS  | = *****     | 7-99      7-100      7-100      7-101      7-101      7-102      7-103                              |
| S.BRA   | = 000740 RG | 7-103   |
| S.CTIM  | = 000714 RG | #11-337      #11-330  |
| UICBUF  | = ***** GX  | *9-174      9-175      *9-177      *9-204      *9-205      9-207      9-209      *9-217      *9-221 |
| USRUIC  | = 000162 RG | #9-200  |
| USRUI1  | = 000154 RG | #9-199  |
| \$CEACX | = ***** GX  | 10-279      10-305  |
| \$CTIM  | = ***** GX  | *11-333   |
| \$ERRYN | = 000222 R  | #7-103      11-335  |
| \$ERRYV | = 000164 R  | #7-102      10-320  |
| \$ERRYX | = 000132 R  | #7-101      10-323  |
| \$ERRYV | = 000050 R  | #7-100      10-321  |
| \$ERRYZ | = 000012 R  | #7-99      10-254   |
| \$HEADR | = ***** GX  | 10-274  |
| \$LINX  | = ***** GX  | 10-311  |
| \$LSTAD | = ***** GX  | *10-300      12-358      *12-372  |



## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE       | REFERENCES  |
|----------|-------------|---|
| CERR     | = ***** GX  | 7-99 7-100 7-101 7-102 7-103 7-104                    |
| CFLIN    | = ***** GX  | 7-99 7-100 7-101 7-102 7-104                          |
| FLAG     | = 000001 R  | #6-86 *8-151 *8-162 8-198                             |
| FMT8     | = ***** GX  | 7-99 7-100 7-101 7-102 7-103 7-104                    |
| FM.E     | = 000000    | #7-99 #7-100 #7-101 #7-102 #7-103 #7-104              |
| KSAF     | = ***** GX  | 8-190 8-214   |
| MXCF     | = 000006 R  | #6-88 *8-120 *8-176 8-199                             |
| N\$SVCT  | = *****     | 8-188   |
| OBJHD    | = ***** GX  | 8-200   |
| ONAME    | = 000002 R  | #6-87 *8-118 *8-119 *8-136 *8-140 8-196 8-197         |
| O.PPY    | = 000202 RG | #8-176  |
| O.END    | = 000224 RG | #8-183  |
| C.FLAG   | = 000146 RG | #8-162  |
| O.FLG    | = 000003    | *8-198  |
| O.LEN    | = 000012    | 8-183   |
| O.MXC    | = 000004    | *8-199  |
| O.NAM    | = 000006    | *8-196 *8-197   |
| O.NAME   | = 000046 RG | #8-133  |
| O.NUM    | = 000000 RG | #8-117  |
| O.TYP    | = 000002    | *8-195 8-209  |
| O.VFY    | = 000120 RG | #8-151  |
| RTSPC    | = ***** GX  | 7-99 7-100 7-101 7-102 7-103 7-104                    |
| \$B\$BAS | = *****     | 7-99 7-100 7-101 7-102 7-103 7-104 7-102 7-102 7-103  |
| TYPE     | = 000000 R  | 7-103 7-104 8-195                                     |
| \$CAT5   | = ***** GX  | #6-85 *8-117 8-209                                    |
| \$CEACX  | = ***** GX  | 8-135 8-139   |
| \$ERRXW  | = 000010 R  | 8-193 8-207   |
| \$ERRXX  | = 000046 R  | #7-99 8-129   |
| \$ERRXY  | = 000110 R  | #7-100 8-147  |
| \$ERRXZ  | = 000154 R  | #7-101 8-158  |
| \$ERRYY  | = 000224 R  | #7-102 8-171  |
| \$ERRYZ  | = 000306 R  | #7-103 8-223  |
| \$HEADR  | = ***** GX  | #7-104 8-222  |
| \$LINFX  | = ***** GX  | 8-188   |
| \$XALOC  | = ***** GX  | 8-213   |
| .PNUMB   | = ***** GX  | 8-186   |
| .PNUMH   | = ***** GX  | 8-117 8-123 8-151 8-152 8-162 8-163 8-165 8-176 8-180 |
| .PSTCN   | = ***** GX  | 8-121   |
| .PSTPT   | = ***** GX  | 8-141 8-133   |

```

314 .SBTTL TPARS STATE TABLES - ROU$DF
315
316 : ROUTING DEFINITION (ROU$DF)
317 :
318
319 000636 ISTAT$ CFGROU,ROUKW
320
321 000636 STATES$ ROU$DF
322 000636 TRANS$ %ROU$DF%,1,SYNERR
323 000636 STATES$ ; MAXIMUM NODE ADDRESS
324 000636 TRANS$ $NUMBR,,R.ADDR
325 000636 STATES$
326 000636 TRANS$ <','>
327 000636 STATES$ ; MAXIMUM COST
328 000636 TRANS$ $NUMBR,,R.COST
329 000636 STATES$
330 000636 TRANS$ <','>
331 000636 STATES$ ; MAXIMUM NUMBER OF HOPS
332 000636 TRANS$ $NUMBR,,R.HOP
333 000636 STATES$
334 000636 TRANS$ <','>
335 000636 STATES$ ; ROUTING TIMER INTERVAL
336 000636 TRANS$ $NUMBR,,R.RTM
337 000636 STATES$
338 000636 TRANS$ <','>
339 000636 STATES$
340 000636 TRANS$ $NUMBR,,R.MAXA ; NUMBER OF AREAS
341 000636 STATES$
342 000636 TRANS$ <','>
343 000636 STATES$
344 000636 TRANS$ $NUMBR,,R.BEA ; NUMBER OF BROADCAST ENDNODES ADJACENT
345 000636 STATES$
346 000636 TRANS$ <','>
347 000636 STATES$
348 000636 TRANS$ $NUMBR,,R.AMXH ; AREA MAXIMUM HOPS
349 000636 STATES$
350 000636 TRANS$ <','>
351 000636 STATES$
352 000636 TRANS$ $NUMBR,,R.AMXC ; AREA MAXIMUM COST
353 000636 STATES$
354 000636 TRANS$ <','>
355 000636 STATES$
356 000636 TRANS$ $NUMBR,$EXIT,R.BTM ; BROADCAST TIMER
357
358
359 000636 STATES$
360
361 000001 .END

```

CF2AC4 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:11 <sup>6 7</sup> Page 9-1  
ACTION ROUTINES TO SET UP EVENT FILTER CONTROL BLOCK

```
218 000356          RETURN
219          ;
220          ; ERROR CONDITIONS
221          ;
222 000360          101$:  EMSG$R  E3          ; FILTER CONTROL BLOCK ALLOCATION FAILURE
223          ;
224          .DSABL  LSB
225          ;
226          000001          .END
```

CF2AC4  
MOVBL

```

217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
000216 010346
000220 012700 000000G
000224 012701 000000G
000230
000234 005020
000236 105020
000237 005001
000238 116701 000000G
000240 012702 000000G
000241 005003
000242 060103
000243
000244 116701 000000G
000245 012702 000000G
000246 060103
000247
000248
000249 116701 000000G
000250 012702 000000G
000251 060103
000252
000253 062703 000000G
000254 010367 000000G
000255 012603
000300
  
```

;+  
 ; SETBLK - SET LOCAL FILE NAME DESCRIPTOR BLOCK  
 ;  
 ; INPUTS -  
 ; DEVBUF - DEVICE STRING  
 ; UICBUF - UIC STRING  
 ; FILBUF - FILE NAME STRING  
 ;  
 ; OUTPUTS -  
 ; FNBUFF - SET UP IN A FILE NAME BLOCK FORMAT  
 ;-  
 SETBLK: MOV R3, -(SP) ; SAVE R3  
 MOV #FNBUFF, R0 ; GET BUFFER ADDRESS  
 MOV #FNBSZ, R1 ; GET BYTE SIZE  
 CALL INIBUF ; INIT LOCAL FNB BUFFER  
 CLR (R0)+ ; ZERO NEXT BLOCK POINTER  
 CLRB (R0)+ ; ZERO USAGE COUNT  
 CLR R1 ; INIT BYTE COUNT  
 MOVB DEVBUF, R1 ; GET FIELD SIZE  
 MOV #DEVBUF, R2 ; GET FIELD ADDRESS  
 CLR R3 ; INIT TOTAL STRING SIZE  
 ADD R1, R3 ; DETERMINE FNB SIZE  
 CALL MOVBUF ; MOVE DEVBUF FIELD  
 MOVB UICBUF, R1 ; GET FIELD SIZE  
 MOV #UICBUF, R2 ; GET FIELD ADDRESS  
 ADD R1, R3 ; UPDATE FNB SIZE  
 CALL MOVBUF ; MOVE NEXT FIELD  
 MOVB FILBUF, R1 ; GET FIELD SIZE  
 MOV #FILBUF, R2 ; GET FIELD ADDRESS  
 ADD R1, R3 ; UPDATE FNB SIZE  
 CALL MOVBUF ; MOVE NEXT FIELD  
 ADD #FNBEXT, R3 ; INCLUDE FNB LINK WORD, USAGE AND LEN BYTES  
 MOV R3, FNBLN ; STORE FNB LENGTH  
 MOV (SP)+, R3 ; RESTORE R3  
 RETURN

CF2AC6 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:11  
Table of contents

|    |     |                   |
|----|-----|-------------------|
| 5- | 53  | MACRO DEFINITIONS |
| 6- | 78  | LOCAL DATA        |
| 7- | 102 | ERROR MESSAGES    |

CF2A

SYME

SYME

CERR

CFL

CTL

DECF

D\$HC

D\$LI

D\$LM

D\$LM

FMT8

FM.8

HSTA

HSTN

I\$8

KSAP

LF.

LF.

LF.

LF.

LF.

LF.

LF.

LF.

LF.

LF.

LF.

LF.

LF.

LGS

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

LN.

CF2AC6      CREATED BY    MACRO    ON 29-JUN-85 AT 00:12      PAGE 1      C 10  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|---|
| CFERR   | = ***** GX  | 7-110      7-111      7-112      7-113      7-114      7-115            |
| CFLIN   | = ***** GX  | 7-110      7-111      7-112      7-113                                  |
| CTL     | = 000005 R  | *6-89      *10-314      10-326  |
| DEPT    | = ***** GX  | 8-130      8-149      8-187      8-245      8-251      8-260      8-267 |
| D\$HOST | = 000022    | *8-261      *8-268  |
| D\$LID  | = 000020    | 8-150      8-188  |
| D\$LNAM | = 000006    | 8-131   |
| D\$LNUM | = 000014    | *8-246      *8-252  |
| FMTR    | = ***** GX  | 7-110      7-111      7-112      7-113      7-114      7-115            |
| FM.8    | = 000000    | #7-110      #7-111      #7-112      #7-113      #7-114      #7-115      |
| HSTARE  | = 000454 RG | #8-256  |
| HSTNUM  | = 000526 RG | #8-265  |
| ISSAS   | = *****     | 8-244      8-259  |
| KSAR5   | = ***** GX  | 8-147      8-170      8-186      8-222      8-231                       |
| LF.ACT  | = 100000    | #5-63   |
| LF.BRC  | = 000400    | #5-63      10-343   |
| LF.BWT  | = 000007    | #5-63   |
| LF.ENA  | = 002000    | #5-63      10-345      10-358   |
| LF.LPB  | = 001000    | #5-63   |
| LF.MDC  | = 000100    | #5-63   |
| LF.MFL  | = 004000    | #5-63      10-356      10-377   |
| LF.MTP  | = 000020    | #5-63   |
| LF.PAC  | = 000200    | #5-63   |
| LF.RDY  | = 040000    | #5-63   |
| LF.REA  | = 010000    | #5-63   |
| LF.SER  | = 000040    | #5-63      10-350   |
| LF.TIM  | = 000010    | #5-63   |
| LF.UNL  | = 020000    | #5-63   |
| LF.X2P  | = 000000    | #5-63   |
| LGSTT   | = ***** GX  | 11-393  |
| LN.CLO  | = 000000    | #5-63   |
| LN.DUM  | = 000005    | #5-63   |
| LN.LOA  | = 000004    | #5-63   |
| LN.LOO  | = 000003    | #5-63   |
| LN.OAU  | = 000003    | #5-63   |
| LN.OFF  | = 000001    | #5-63   |
| LN.ON   | = 000000    | #5-63   |
| LN.ONP  | = 000004    | #5-63   |
| LN.OPE  | = 000001    | #5-63   |
| LN.REF  | = 000002    | #5-63   |
| LN.SER  | = 000002    | #5-63   |
| LN.STA  | = 000017    | #5-63   |
| LN.SUB  | = 000360    | #5-63   |
| LN.TRI  | = 000006    | #5-63   |
| L.COST  | = 000015    | #5-63   |
| L.CTL   | = 000012    | #5-63      10-326   |
| L.CVA   | = 177776    | #5-63   |
| L.DDM   | = 000002    | #5-63      10-330   |
| L.DDS   | = 000004    | #5-63   |
| L.DLC   | = 000003    | #5-63   |
| L.DLM   | = 000006    | #5-63   |
| L.DLS   | = 000010    | #5-63   |

```

331 ;+
332 ; CRSUB - CREATE A SUB-PARTITION
333 ;
334 ; INPUTS:
335 ; R0 = SUB-PCB ADDRESS
336 ; R5 = MAIN PCB ADDRESS
337 ;
338 ; OUTPUTS:
339 ; C-BIT = SUCCESS/FAILURE
340 ; R0,R1,R2,R3,R4 = DESTROYED
341 ; R5 = SUB-PCB ADDRESS
342 ; -
343 ;
344 000736 010001 CRSUB: .IF DF R$$11M & M$$MGE
345 000740 016702 MOV R0,R1 ;* COPY PCB ADDRESS
346 000744 006202 MOV PLGTH,R2 ;* GET PCB LENGTH
347 000746 005021 ASR R2 ;* CONVERT TO A WORD COUNT
348 000750 5$: CLR (R1)+ ;* ZERO THE ENTIRE SUB-PCB
349 000754 016760 SOB R2,5$ ;*
350 000762 016760 MOV .PNAM1,P.NAM(R0) ;* SET PARTITION NAME
351 000770 010550 MOV .PNAM1+2,P.NAM+2(R0) ;* FROM FIRST PARTITION IN PAR$DF
352 000774 016760 MOV R5,P.MAIN(R0) ;* MAIN PCB ADDRESS
353 001002 066760 MOV .NTPSZ,P.BLKS(R0) ;* SIZE IN 32. WORD BLOCKS
354 ; ADD .FUDGE,P.BLKS(R0) ;*
355 ;
356 ; .IF DF R$$MPL
357 ;
358 ; MOV #PS.COM!PS.NSF!PS.FXD!PS.CHK,P.STAT(R0) ;* COMMON, NO
359 ; ;* SHUFFLING, FIXED, NO CHECKPTING
360 ; MOV #P2.ROn,P.ST2(R0) ;* MAKE REGION READ-ONLY
361 ;
362 ; .IFF
363 0G1010 012760 000000C 000000G MOV #PS.COM!PS.NSF!PS.SYS,P.STAT(R0) ;* COMMON, NO SHUFFLING
364 ;
365 ; .ENDC
366 ;
367 001016 010501 MOV R5,R1 ;* COPY MAIN PCB ADDRESS
368 001020 005002 CLR R2 ;* NO PREVIOUS SUB-PCB
369 001024 005046 CLR ~(SP) ;* PREVIOUS SUB-PCB IF SPACE IS FOUND
370 001026 066102 10$: ADD P.REL(R1),R2 ;* GIVING BIAS OF NEXT HOLE IN PARTITION
371 001030 016103 MOV P.SUB(R1),R3 ;* GET NEXT SUB-PCB
372 001034 001425 BEQ 30$ ;* IF ZERO, END OF LIST
373 001036 016304 MOV P.REL(R3),R4 ;* GET BIAS OF SUB-PARTITION
374 001042 160204 SUB R2,R4 ;* GIVING SIZE OF HOLE
375 001044 166004 SUB P.BLKS(R0),R4 ;* IS THE HOLE BIG ENOUGH?
376 001050 103413 BLO 20$ ;* IF M1, NO
377 001052 010116 MOV R1,(SP) ;* SAVE PREVIOUS SUB-PCB ADDRESS
378 001054 010260 MOV R2,P.REL(R0) ;* SET SUB PARTITION BIAS
379 001060 010360 MOV R3,P.SUB(R0) ;* AND POINTER TO NEXT SUB-PCB IN LIST
380 001064 032767 BIT #LS.TOP,$FUNC ;* TOP-DOWN ALLOCATION?
381 001072 001450 BEQ 50$ ;* IF EQ, NO - ALL DONE
382 001074 060460 ADD R4,P.REL(R0) ;* ALLOCATE FROM TOP OF HOLE
383 001100 010301 20$: MOV R3,R1 ;* COPY SUB-PCB ADDRESS
384 001102 016102 MOV P.BLKS(R1),R2 ;* GET SUB-PCB SIZE
385 001106 000746 BR 10$ ;*
386 001110 016504 30$: MOV P.REL(R5),R4 ;* GET MAIN PARTITION BIAS
387 001114 066504 ADD P.BLKS(R5),R4 ;* GIVING BIAS OF END OF PARTITION

```

```

124
125
126          ;+ SERINI - INIT SERVICE DATA BASE BUFFERS
127          ;
128          ; INPUTS -
129          ;     NONE
130          ;
131          ; OUTPUTS -
132          ;     SER$DF BUFFERS INITIALIZED
133          ; -
134
135 000056 012700 000000G  SERINI::MOV    #SERBEG,R0    ; GET START OF BUFFERS
136 000062 005020          S$:    CLR    (R0)+      ; CLEAR ONE WORD
137 000064 02 700 000000G    CMP    #SERFIN,R0    ; CHECK FOR END
138 000070 001374          BNE    S$      ; IF MORE - BRANCH
139
140 000072          RETURN
141
142
143

```



CF2AC8      CREATED BY    MACRO    ON 29-JUN-85 AT 00:12      PAGE 5      C 13

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|          |        |       |       |       |       |       |       |       |        |        |
|----------|--------|-------|-------|-------|-------|-------|-------|-------|--------|--------|
| CALL     | 6-107  | 6-108 | 6-116 | 8-159 | 8-167 | 8-178 | 8-186 | 8-188 | 8-199  | 8-238  |
|          | 8-244  | 9-285 | 9-287 |       |       |       |       |       |        |        |
| DHBD\$   | #5-50  | 5-58  |       |       |       |       |       |       |        |        |
| EMSG\$R  | #5-50  | 8-183 |       |       |       |       |       |       |        |        |
| FNBD\$   | #5-50  | 5-59  |       |       |       |       |       |       |        |        |
| NTLR\$   | #5-50  | 5-85  |       |       |       |       |       |       |        |        |
| PDVDF\$  | #5-51  | 5-61  |       |       |       |       |       |       |        |        |
| RESMAP   | #5-69  | 8-245 |       |       |       |       |       |       |        |        |
| RETC     | #5-50  | 8-191 |       |       |       |       |       |       |        |        |
| RETURN   | 6-122  | 7-140 | 8-174 | 8-181 | 8-192 | 8-247 | 9-301 | 9-314 | 10-335 | 10-366 |
|          | 11-413 |       |       |       |       |       |       |       |        |        |
| SAVMAP   | #5-65  | 8-104 |       |       |       |       |       |       |        |        |
| SERDF\$  | #5-50  | 5-60  |       |       |       |       |       |       |        |        |
| SLTDF\$  | #5-51  | 5-62  |       |       |       |       |       |       |        |        |
| SWST. \$ | 8-186  |       |       |       |       |       |       |       |        |        |

CF2AC9 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:13 Page 9-1  
SER\$DF ACTION ROUTINES

```
379  
380 001026 016767 000000G 000000G YPCIR:: MOV .PNUMB,XPTLLN ; STORE XPT CIRCUIT COUNT  
381 001034 RETURN  
382  
383 001036 016767 000000G 000000G ECLML:: MOV .PNUMB,ECLLLN ; STORE MAX LINKS VALUE  
384 001044 RETURN  
385  
386 001046 016767 000000G 000000G ECLMC:: MOV .PNUMB,ECLNOD ; STORE MAX NODE COUNTERS VALUE  
387 001054 RETURN  
388  
389 001056 066767 000000G 000000G BRDJ:: ADD .PNUMB,.BRAD ; CALCULATE BROADCAST ROUTER ADJACENCIES  
390 001064 RETURN  
391
```

FILE ID CF2DEC

C 15

```
CCCCCCCC FFFFFFFF 222222 DDDDDDDD EEEEEEEEE CCCCCCCC
CCCCCCCC FFFFFFFF 222222 DDDDDDDD EEEEEEEEE CCCCCCCC
CC        FF      22      DD      EE      CC
CC        FF      22      DD      EE      CC
CC        FF      22      DD      EE      CC
CC        FF      22      DD      EE      CC
CC        FFFFFFFF 22      DD      EEEEEEEE CC
CC        FFFFFFFF 22      DD      EEEEEEEE CC
CC        FF      22      DD      EE      CC
CC        FF      22      DD      EE      CC
CC        FF      22      DD      EE      CC
CC        FF      22      DD      EE      CC
CCCCCCCC FF 222222222 DDDDDDDD EEEEEEEEE CCCCCCCC
CCCCCCCC FF 222222222 DDDDDDDD EEEEEEEEE CCCCCCCC
```

```
LL          SSSSSSSS TTTTTTTTTT
LL          SSSSSSSS TTTTTTTTTT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LL          SSSSSS  TT
LL          SSSSSS  TT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

\*\*FILE\*\*ID\*\*CF2UMR

```

CCCCCCCC  FF,FFFFFFF  222222  UU      UU  MM      MM  RRRRRRRR
CCCCCCCC  FFFFFFFF    222222  UU      UU  MM      MM  RRRRRRRR
CC        FF          22      22  UU      UU  MMMM   MMMM  RR      RR
CC        FF          22      22  UU      UU  MMMM   MMMM  RR      RR
CC        FF          22      22  UU      UU  MM      MM  RR      RR
CC        FFFFFFFF    22      22  UU      UU  MM      MM  RRRRRRRR
CC        FFFFFFFF    22      22  UU      UU  MM      MM  RRRRRRRR
CC        FF          22      22  UU      UU  MM      MM  RR      RR
CC        FF          22      22  UU      UU  MM      MM  RR      RR
CC        FF          22      22  UU      UU  MM      MM  RR      RR
CC        FF          22      22  UU      UU  MM      MM  RR      RR
CCCCCCCC  FF          2222222222  UUUUUUUUUU  MM      MM  RR      RR
CCCCCCCC  FF          2222222222  UUUUUUUUUU  MM      MM  RR      RR

```

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LL          SSSSSS    TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT

```

.SBTTL X2P\$DF ACTION ROUTINES

```

195
196
197
198
199
200
201 000216 022767 000003 000000G X2.DEV: .ENABL LSB
202 000224 103464 000000G X2.DEV: CMP #3, PSTCN ; VALID DEVICE NAME?
203 000226 016700 000000G X2.DEV: BLO 101$ ; BR IF NO
204 000232 010167 177536 X2.DEV: MOV .PSTPT, R0 ; POINT TO START OF NAME
205 000236 010167 177536 X2.DEV: CALL $CAT5 ; CONVERT TO RAD50
206 000242 010167 177536 X2.DEV: MOV R1, X2DEV ; STORE DEVICE NAME
207
208
209
210 000244 005767 000000G X2.CTL: TST .PNUMH ; LEGAL CONTROLLER NUMBER?
211 000250 001052 000000G X2.CTL: BNE 101$ ; BR IF NO
212 000252 105767 000001G X2.CTL: TSTB .PNUMB+1 ; LEGAL?
213 000256 001047 000000G X2.CTL: BNE 101$ ; BR IF NO
214 000260 116767 000000G 177514 X2.CTL: MOVB .PNUMB, X2CTL ; SAVE CONTROLLER NUMBER
215 000266 105067 177511 X2.CTL: CLRB X2UNT ; DEFAULT UNIT TO ZERO
216 000272
217
218
219
220 000274 005767 000000G X2.UNIT: TST .PNUMH ; LEGAL UNIT NUMBER?
221 000300 001036 000000G X2.UNIT: BNE 101$ ; BR IF NO
222 000302 105767 000001G X2.UNIT: TSTB .PNUMB+1 ; LEGAL?
223 000306 001033 000000G 177465 X2.UNIT: BNE 101$ ; BR IF NO
224 000310 116767 000000G 177465 X2.UNIT: MOVB .PNUMB, X2UNT ; SAVE UNIT NUMBER
225 000316 016700 000000G CHKDEV: MOV $SLTA, R0 ; GET SLT ADDRESS
226 000322 116001 000002 X2.UNIT: MOV L.DDM(R0), R1 ; GET DDM'S PDV INDEX
227 000326 067701 000000G X2.UNIT: ADD @PDVTA, R1 ; INDEX INTO PDV VECTOR TABLE
228 000332 011101 000000G X2.UNIT: MOV (R1), R1 ; GET PDV INDEX
229 000334 026167 000004 177436 X2.UNIT: CMP Z.NAM(R1), X2DEV ; IS THIS THE CORRECT DEVICE?
230 000342 001010 000000G X2.UNIT: BNE 10$ ; BR IF NO
231 000344 126067 000012 177430 X2.UNIT: CMPB L.CTL(R0), X2CTL ; IS THIS THE CORRECT CONTROLLER?
232 000352 001004 000000G X2.UNIT: BNE 10$ ; BR IF NO
233 000354 126767 000000G 177421 X2.UNIT: CMPB UCNT, X2UNT ; IS THIS THE CORRECT UNIT NUMBER?
234 000362 001404 000000G X2.UNIT: BEQ 20$ ; BR IF YES - FOUND IT
235 000364
236 000374
237
238 000376 101$: MSG$R 2I ; ILLEGAL LINE-ID
239
240
241
242
243
244 000404 005767 000000G X2.MWS: TST .PNUMH ; LEGAL MAXIMUM WINDOW SIZE VALUE?
245 000410 001016 000000G X2.MWS: BNE 101$ ; BR IF NO
246 000412 016700 000000G X2.MWS: MOV .PNUMB, R0 ; GET MAXIMUM WINDOW SIZE
247 000416 022700 000007 X2.MWS: CMP #X2WNMX, R0 ; IS IT VALID?
248 000422 103411 000001 X2.MWS: BLO 101$ ; BR IF NO
249 000424 022700 000001 X2.MWS: CMP #X2WNMN, R0 ; ...
250 000430 101006 000000G X2.MWS: BHI 101$ ; BR IF NO
251 000432 016701 000000G X2.MWS: MOV UCNT, R1 ; GET UNIT COUNT
252 000436 006301 000000G X2.MWS: ASL R1 ; MAKE IT A WORD INDEX

```

```
62          .SBTTL  CONSTANT DEFINITIONS
63
64          :
65          : CONSTANT DEFINITIONS
66          :
67          000040          BLKSMN = 32.          ; MINIMUM BLOCK SIZE VALUE
68          002000          BLKSMX = 1024.        ; MAXIMUM BLOCK SIZE VALUE
69          000177          WNDSMX = 127.         ; MAXIMUM WINDOW SIZE VALUE
70          000007          X3TMCT = 7           ; NUMBER OF TIMER PARAMETERS IN X3P$DF
```

.TITLE CF2AC1 - INITIAL CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NLT - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 14-DEC-79  
DECnet-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECnet-11M V3.1  
DECNET-11M-PLUS V1.1
- 3.01 11-JAN-83  
SUPPORT CHANGED CETAB PARAMETERS
- 4.00 07-NOV-83  
DECnet-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0

CF2AC1      CREATED BY    MACRO    ON 29-JUN-85 AT 00:09      PAGE 2      D 4

SYMBOL CROSS REFERENCE      CREF    04.00

SYMBOL    VALUE      REFERENCES

|        |            |         |        |         |        |        |        |        |
|--------|------------|---------|--------|---------|--------|--------|--------|--------|
| SLSTRM | = ***** GX | *10-299 | 12-362 | *12-371 |        |        |        |        |
| EXALOC | = ***** GX | 10-272  |        |         |        |        |        |        |
| .PNUM2 | = ***** GX | 10-256  | 10-258 | 10-267  | 10-269 | 11-333 | 11-337 | 11-339 |
| .PNUM1 | = ***** GX | 11-331  |        |         |        |        |        |        |
| .PSTCN | = ***** GX | *8-126  | 10-242 | *10-247 |        |        |        |        |
| .PSTPT | = ***** GX | 8-122   | 10-245 |         |        |        |        |        |

CF2A  
MACR  
MACR  
CALL  
EMSG  
NTLE  
OBJD  
RESM  
RETC  
RETC  
SAVM  
SWST



CF2AC2      CREATED BY    MACRO    ON 29-JUN-85 AT 00:10      PAGE 2      D 5

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|
| CALL    | 8-135 | 8-139 | 8-185 | 8-186 | 8-193 | 8-207 | 8-213 |
| EMSG\$R | #5-60 | 8-129 | 8-147 | 8-158 | 8-171 | 8-222 | 8-223 |
| NTLR\$  | #5-60 | 7-99  | 7-100 | 7-101 | 7-102 | 7-103 | 7-104 |
| OBJDF\$ | #5-60 | 5-62  |       |       |       |       |       |
| RESMAP  | #5-71 | 8-214 |       |       |       |       |       |
| RET     | #5-60 | 8-188 |       |       |       |       |       |
| RETURN  | 8-125 | 8-143 | 8-154 | 8-167 | 8-215 | 8-218 |       |
| SAVMAP  | #5-67 | 8-190 |       |       |       |       |       |
| SWSTK\$ | 8-185 |       |       |       |       |       |       |

```

A$$CHK= 000000      F$$LVL= 000001      N$$NCT= 000001      X$$DBT= 000000      $$SUBXP= 000010
A$$CPS= 000000      G$$TPP= 000000      N$$PEM= 000001      $ALPHA= 000022      $$$FLG= 177777
A$$PRI= 000000      G$$TSS= 000000      P$$P45= 000000      $ANY = 000020      $$$KEY= 000000
A$$TRP= 000000      G$$TTK= 000000      P$$WRD= 000000      $BLANK= 000006      $$$STA= 000000
BFEND = 000040R      002 G$$WRD= 000000      QROU = 000224RG      $DIGIT= 000024      $$$TMP= 000000R      005
BFTAB = 000000R      002 I$$RAR= 000000      Q$$OPT= 000010      $DNUM= 000014      .AMAXC= ***** GX
B.END = 000060RG      I$$RDN= 000000      ROUDF = 000000R      003 $EOS = 000012      .AMAXH= ***** GX
B.NUM = 000014RG      K$$CNT= 177546      ROUKW = 000000RG      004 $ERRXR= 000040R      .BFPTR= ***** GX
B.STR = 000000RG      K$$CSR= 177546      RTSPC = ***** GX      $ERRXS= 000074R      .CCBNM= ***** GX
B.ZEND= 000054RG      K$$LDC= 000000      R$DER= 000000      $ERRYI= 000632R      .CCBSZ= ***** GX
B.ZERO= 000010RG      K$$TPS= 000074      R$K11= 000001      $ERRYJ= 000700R      .MAXC = ***** GX
CERR = ***** GX      LD$LP = 000000      R$SND= 000000      $ERRYK= 000126R      .MAXH = ***** GX
CFGBF = ***** GX      L$ASG= 000000      R$11M= 000000      $ERRYL= 000162R      .MXPRO= ***** GX
CFGROU= 000000RG      003 L$DRV= 000000      R.ADDR= 000276RG      $ERRYM= 000214R      .NA = ***** GX
CFGSZ = ***** GX      L$P11= 000001      R.AMXH= 000544RG      $ERRYN= 000240R      .NBEA = ***** GX
CFLIN = ***** GX      L$11R= 000000      R.BEA = 000452RG      $ERYO = 000342R      .NN = ***** GX
CRCOV= ***** GX      M$CRB= 000124      R.BTM = 000572RG      $ERRYP= 000424R      .NTPSZ= ***** GX
C$CKP= 000000      M$CRX= 000000      R.COST= 000306RG      $ERRYQ= 000464R      .PNUMB= ***** GX
C$ORE= 000400      M$FCS= 000000      R.HOP = 000334RG      $ERRYV= 000524R      .PNUMH= ***** GX
C$SRSH= 177564      M$MGE= 000000      R.MAXA= 000406RG      $ERRYW= 000562R      .RDBNM= ***** GX
D$BUG= 177514      M$NET= 000000      R.RTM = 000362RG      $ERRIT= ***** GX      .RDBSZ= ***** GX
D$ISK= 000000      M$OVR= 000000      R.RTM = 000362RG      $EXIT = 000000      .RDBTH= ***** GX
D$SL11= 000001      N$ACC= 000001      SYNERR= ***** GX      $FAIL = 177777      .ROUTB= ***** GX
D$SYNC= 000000      N$BUF= 000001      S$WRG= 000000      $LAMD= 000000      .ROUTM= ***** GX
D$SYNM= 000000      N$LDV= 000001      S$YSZ= 007600      $MUL = ***** GX      .SDBNM= ***** GX
E$XPR= 000000      N$MCP= 000001      T$KMG= 000000      $NUMBER= 000002      .SDBSZ= ***** GX
FMT8 = ***** GX      N$MLL= 000001      T$MIN= 000000      $RAD50= 000016      .TPARS= ***** GX
FM.8 = 000000      N$MOV= 000010      V$CTR= 001000      $STRNG= 000004

```

```

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)
        000636 001 (RW,I,LCL,REL,CON)
DATA    000746 002 (RW,D,LCL,REL,CON)
$STATE 000074 003 (RW,D,LCL,REL,CON)
$KTAB  000002 004 (RW,D,LCL,REL,CON)
$KSTR  000007 005 (RW,D,LCL,REL,CON)
Errors detected: 0

```

### \*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 12656 Words ( 50 Pages)
Size of core pool: 14440 Words ( 55 Pages)
Operating system: RSX-11M'PLUS

```

Elapsed time: 00:00:36.18  
SY:CF2AC3.V2,[132,134]CF2AC3/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130, .UJNETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CF2AC3

Symbol table

|                 |                  |                   |                  |                     |
|-----------------|------------------|-------------------|------------------|---------------------|
| ASSCHK= 000000  | FF.ADD= 000020   | F.EVT 000004      | L\$DRV= 000000   | R\$DER= 000000      |
| ASSCPS= 000000  | FF.CIR= 000040   | F.FLG 000014      | L\$P11= 000001   | R\$K11= 000001      |
| ASSPRI= 000000  | FF.CON= 000001   | F.LEN 000016      | L\$11R= 000000   | R\$SND= 000000      |
| ASSTRP= 000000  | FF.FIL= 000002   | F.LIN 000020      | M\$CRB= 000124   | R\$11M= 000000      |
| CERR = ***** GX | FF.HST= 040000   | F.LNK 000000      | M\$CRX= 000000   | STRADD 000014R 002  |
| CLASS 000000R   | FF.LIN= 000010   | F.MOD 000022      | M\$FCS= 000000   | S\$WRG= 000000      |
| C\$CKP= 000000  | FF.MOD= 000100   | F.REM 000026      | M\$MGE= 000000   | S\$YSZ= 007600      |
| C\$ORE= 000400  | FF.MON= 000004   | F.SEV 000004      | M\$NET= 000000   | T\$KMG= 000000      |
| C\$RSH= 177564  | FF.MSK= 000077   | G\$TPP= 000000    | M\$OVR= 000000   | T\$MIN= 000000      |
| D\$BUG= 177514  | FF.PRT= 000200   | G\$TSS= 000000    | N\$ACC= 000001   | V\$CTR= 001000      |
| D\$ISK= 000000  | FF.QL= 000370    | G\$TK= 000000     | N\$BUF= 000001   | X\$DBT= 000000      |
| D\$LL1= 000001  | FF.REM= 100000   | G\$WRD= 000000    | N\$LDV= 000001   | \$CEACX= ***** GX   |
| D\$SYNC= 000000 | FILHD = ***** GX | I\$RAR= 000000    | N\$MCP= 000001   | \$ERRE1 000016R 002 |
| D\$SYNM= 000000 | FLAG 000002R 002 | I\$RDN= 000000    | N\$MLL= 000001   | \$ERRE2 000052R 002 |
| EVENT 000004R   | FMT8 = ***** GX  | K\$AR5 = ***** GX | N\$MDV= 000010   | \$ERRE3 000120R 002 |
| E\$XPR= 000000  | FM.8 = 000000    | K\$CNT= 177546    | N\$NCT= 000001   | \$ERRE4 000206R 002 |
| E.CON 000130RG  | F\$LVL= 000001   | K\$CSR= 177546    | N\$PEM= 000001   | \$HEADR= ***** GX   |
| E.FIL 000140RG  | F.ADD 000016     | K\$LDC= 000000    | P\$P45= 000000   | \$LINKX= ***** GX   |
| E.MON 000150RG  | F.CEV 000030     | K\$TPS= 000074    | P\$WRD= 000000   | \$XALOC= ***** GX   |
| E.MSK 000060RG  | F.CIR 000024     | LD\$LP = 000000   | Q\$OPT= 000010   | .PNUMB= ***** GX    |
| E.NUM 000000RG  | F.CLS 000002     | L\$ASG= 000000    | RTSPC = ***** GX | .PNUMH= ***** GX    |

. ABS. 000040 000 (RW,I,GBL,ABS,OVR)  
 000566 001 (RW,I,LCL,REL,CON)  
 DATA 000256 002 (RW,D,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 11101 Words ( 44 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:11.10  
 SY:CF2AC4.V2,[132,134]CF2AC4/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CF2AC4

```

259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315

```

```

.SBTTL CRFNBK - CREATE FILENAME DESCRIPTOR BLOCK

+
CRFNBK - CREATE FILE NAME BLOCK
:
: INPUTS -
:         FNBUFF - LOCAL COPY OF FNB DATA
:         FNBLN  - SIZE OF THIS FNB
:
: OUTPUTS -
:         FILNAME DESCRIPTOR BLOCK CREATED
:
CRFNBK: SWSTK$ 30$      ; ENTER SYSTEM STATE
      MOV     FNBLN,R1  ; GET FNB SIZE
      CALL    $XALOC    ; ALLOCATE FNB
      BCC     10$       ; IF SUCCESS - BRANCH
      RETC    R0        ; IF NOT - SET USER C-BIT
      BR      30$       ; RETURN

10$:  MOV     @KSARS,-(SP) ; SAVE MAPPING
      MOV     R0,R2      ; SAVE UNMAPPED BUFFER ADDRESS
      MOV     R0,-(SP)   ; SET BUFFER ADDRESS

      CALL    $CEACX     ; GET BUFFER ADDRESS
      MOV     (SP)+,R0   ; GET MAPPED ADDRESS

      MOV     FNBLN,R1   ; GET FNB LENGTH
      MOV     R2,-(SP)   ; SAVE UNMAPPED ADDRESS
      MOV     #FNBUFF,R2 ; GET LOCAL COPY OF NEW FNB DATA
      MOVB    (R2)+,(R0)+ ; TRANSFER DATA FROM LOCAL FNB BUFFER
                          ; TO ACTUAL FNB IN POOL..
      DEC     R1         ; COUNT ONE LESS BYTE
      BNE     15$       ; IF MORE - BRANCH

      MOV     (SP)+,R2   ; RESTORE UNMAPPED ADDRESS
      MOV     @DECP1,R0  ; GET DECNET HOME BLOCK ADDRESS
      ADD     #D$FNB,R0  ; GET FNB LISTHEAD
      MOV     R0,R3      ; SAVE LISTHEAD

      MOV     R3,R1      ; STORE CURRENT UNMAPPED FNB ADDRESS
      MOV     (R0),R0    ; GET NEXT FNB
      BEQ     25$       ; IF END OF LIST - INSERT NEW FNB
      MOV     R0,R3      ; SAVE UNMAPPED ADDRESS OF NEW CURRENT FNB
      MOV     R0,-(SP)   ; PUT UNMAPPED ADDRESS ON STACK FOR $CEACX
      CALL    $CEACX     ; MAP TO BUFFER
      MOV     (SP)+,R0   ; GET MAPPED ADDRESS
      BR      20$       ; CONTINUE

20$:  MOV     R3,R1      ; STORE CURRENT UNMAPPED FNB ADDRESS
      MOV     (R0),R0    ; GET NEXT FNB
      BEQ     25$       ; IF END OF LIST - INSERT NEW FNB
      MOV     R0,R3      ; SAVE UNMAPPED ADDRESS OF NEW CURRENT FNB
      MOV     R0,-(SP)   ; PUT UNMAPPED ADDRESS ON STACK FOR $CEACX
      CALL    $CEACX     ; MAP TO BUFFER
      MOV     (SP)+,R0   ; GET MAPPED ADDRESS
      BR      20$       ; CONTINUE

25$:  MOV     R2,4(SP)    ; STORE UNMAPPED ADDRESS OF NEW FNB IN USER R0
      MOV     R2,-(SP)   ; PUT UNMAPPED ADDRESS ON STACK FOR $LINKX
      MOV     R1,-(SP)   ;
      CALL    $LINKX     ; LINK NEW FNB INTO LIST
      MOV     (SP)+,@KSARS ; RESTORE MAPPING

```

.TITLE CF2AC6 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

Copyright (c) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

# MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

## IDENT HISTORY:

- 1.00 12-JAN-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Mic-o/RX V1.0

CF2A  
SYME  
SYME  
L.FU  
L.KF  
L.LB  
L.LC  
L.MF  
L.NF  
L.NF  
L.NF  
L.OV  
L.UN  
MRK  
NAME  
NOD  
NOD  
NOD  
PDV  
PDV  
PRM  
PRM  
RTS  
R\$  
R\$  
SF.  
SF.  
SF.  
SF.  
SF.  
SF.  
SLT  
SLT  
STA  
STR  
STR  
STR  
STR  
STR  
STR  
S\$  
S.C  
S.C  
S.F  
S.L  
S.N  
S.N  
S.O  
S.U  
UNT  
X.F

CF2AC6 CREATED BY MACRO ON 29-JUN-85 AT 00:12 PAGE 2 D 10  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES   |
|---------|------------|--|
| L.FLG   | 000000     | #5-63 *10-342 10-343 10-345 10-350 10-356 *10-358 10-377 |
| L.KRBA  | 000016     | #5-63  |
| L.LEN   | = 000022   | #5-63  |
| L.LGST  | 001232 RG  | #11-589  |
| L.MPF   | 000022     | #5-63 10-347 10-352 10-373                               |
| L.NMST  | 000020     | #5-63  |
| L.NSTA  | 000014     | #5-63 10-366   |
| L.OWNR  | 000021     | #5-63  |
| L.UNT   | 000013     | #5-63 10-328 10-376                                      |
| MRKFL   | = ***** GX | 10-342   |
| NAME    | 000000 R   | #6-88 *9-280 9-288 *10-308 10-334                        |
| NUDARE  | 000356 RG  | #8-241   |
| NODID   | 000024 RG  | #8-140   |
| NODNM   | 000000 RG  | #8-130   |
| NODNUM  | 000430 RG  | #8-249   |
| PDVNM   | = ***** GX | 9-283  |
| PDVTA   | = ***** GX | 9-281 10-331   |
| PRMFL   | 000644 RG  | #9-298   |
| PRONM   | 000560 RG  | #9-277   |
| RTSPC   | = ***** GX | 7-110 7-111 7-112 7-113 7-114 7-115                      |
| RSSEIS  | = *****    | 8-244 8-259  |
| RS\$11D | = *****    | 8-244 8-259  |
| SF.ACT  | = 000200   | #5-63  |
| SF.ENA  | = 000100   | #5-63 10-348 10-374 10-379                               |
| SF.LPB  | = 000004   | #5-63  |
| SF.MFL  | = 000040   | #5-63 10-375   |
| SF.PAC  | = 000020   | #5-63  |
| SF.REA  | = 000010   | #5-63  |
| SF.SER  | = 000001   | #5-63 10-353   |
| SF.SVC  | = 000002   | #5-63  |
| SF.UNL  | = 000040   | #5-63  |
| SLTMA   | = ***** GX | 10-320   |
| SLTNM   | = ***** GX | 10-321   |
| STAN    | = ***** GX | *10-341 10-368 *10-369                                   |
| STRAD   | 000006 R   | #6-96 *8-162 *8-193 8-199                                |
| STRCH   | 000154 RG  | #8-182   |
| STRCT   | 000012 R   | #6-98  |
| STRMX   | 000010 R   | #6-97 *8-164 8-168 8-183 8-205 8-210 8-213 8-220 8-229   |
| STRPT   | 000014 R   | #6-99 *8-172 *8-173 8-201                                |
| STRXX   | 000016 R   | #6-100 8-134   |
| SS\$BAS | = *****    | 7-110 7-111 7-112 7-113 7-114 7-115                      |
| S.COST  | 000001     | #5   |
| S.CTL   | 000700 RG  | #5-314   |
| S.FLG   | 000000     | #5-63 *10-374 *10-375 *10-376 *10-379                    |
| S.LEN   | 000004     | #5-63  |
| S.NAME  | 000660 RG  | #10-305  |
| S.NMS   | 000002     | #5-63  |
| S.OWNR  | 000003     | #5-63  |
| S.UNT   | 000710 RG  | #10-320  |
| UNT     | 000004 R   | #6-90  |
| X.FLG   | 001124 RG  | #10-364  |

```

388 001120 160204          SUB      R2,R4          ;* GIVING SIZE OF LAST HOLE
389 001122 166004          SUB      P,BLK(S(R0),R4 ;* IS LAST HOLE BIG ENOUGH?
390 001126 103412          BLO      50$           ;* IF M1, NO
391 001130 032767          BIT      #LS.TOP,$FUNC ;* TOP-DOWN ALLOCATION?
392 001136 001401          BEQ      40$           ;* IF EQ, NO
393 001140 060402          ADD      R4,R2          ;* ALLOCATE FROM TOP OF HOLE
394 001142 010116          40$: MOV      R1,(SP)      ;* SAVE PREVIOUS SUB-PCB ADDRESS
395 001144 010260          MOV      R2,P.REL(R0)    ;* SET SUB PARTITION BIAS
396 001150 010360          MOV      R3,P.SUB(R0)    ;* AND POINTER TO NEXT SUB-PCB IN LIST
397 001154 012601          50$: MOV      (SP)+,R1    ;* GET PREVIOUS SUB-PCB ADDRESS
398 001156 001405          BEQ      101$           ;* IF EQ, NO SPACE IN PARTITION
399 001160 010061          MOV      R0,P.SUB(R1)    ;* LINK OUR SUB-PCB TO PREVIOUS
400 001164 010005          MOV      R0,R5          ;* THIS IS NOW THE OFFICIAL PCB
401 001166 000241          CLC                     ;* INDICATE SUCCESS
402 001170          RETURN                          ;*
403
404          ;
405          ; ERROR CONDITIONS
406          ;
407 001172 016701          101$: MOV      PLGTH,R1    ;* GET PCB LENGTH
408 001176          CALL      @DEACB                ;* DE-ALLOCATE SUB-PCB
409 001202          SEC                             ;* INDICATE FAILURE
410 001204          RETURN                          ;*
411          .ENDC
412
413
414          000001          .END

```

```

145      ;+
146      ; HRDADD - STORE HARDWARE ADDRESS
147
148      ; INPUTS -
149      ; .PSTPT - PARSED STRING ADDRESS
150      ; .PSTCN - STRING SIZE
151
152      ; OUTPUTS -
153      ; HARDWARE ADDRESS CONVERTED
154      ; -
155
156 000074 010345      HRDADD::MOV      R3, -(SP)      ; SAVE REG
157 000076 016700      MOV      .PSTPT, R0      ; GET PARSED STRING ADDRESS
158 000102 012702 000000G MOV      #SERADD, R2      ; GET HARDWARE ADDRESS BUFFER
159 000106      CALL      HEXIND      ; CONVERT HEX
160
161      ASL      R3      ; POSITION DIGIT
162      ASL      R3      ; ...
163      ASL      R3      ; ...
164      ASL      R3      ; ...
165
166      MOVB     R3, (R2)      ; STORE DIGIT
167      CALL     HEXIND      ; CONVERT HEX
168      BISB     R3, (R2)+      ; STORE DIGIT
169
170 000132 022702 000000G CMP      #SERTMP, R2      ; END OF BUFFER ?
171 000135 001363      BNE      5$,      ; RESTORE REG
172 000140 012603      MOV      (SP)+, R3
173
174      RETURN
175
176
177
178 000144      SEREND::CALL     SERALL      ; ALLOCATE SERVICE BLOCK
179 000150 103401      BCS      101$      ; IF ALLOCATION FAILURE BRANCH
180
181      RETURN
182
183 000152      101$: MSG$R      SS      ; SERVICE BLOCK ALLOCATION FAILURE
184
185
186 000162      SE$ALL: SWSTK$ 30$      ; ENTER SYSTEM STATE
187 000166 012701 000052      MOV      #S LENG, R1      ; GET SIZE OF BLOCK NEEDED
188 000172      CALL      $XALOC      ; ALLOCATE BLOCK FROM POOL..
189 000176 103006      BCC      10$      ; IF SUCCESS - BRANCH
190
191      RETC     R0      ; IF FAILURE - SET USER C-BIT
192      RETURN
193
194 000214      10$: SAVMAP      ; SAVE MAPPING
195 000220 010002      MOV      R0, R2      ; SAVE UNMAPPED ADDRESS
196 000222 010067 000000G MOV      R0, SERCUR      ; SAVE CURRENT SERVICE BLOCK ADDRESS
197
198      MOV      R0, -(SP)      ; SET UP UNMAPPED ADDRESS
199      CALL     $CEACK      ; MAP TO BLOCK
200 000234 012600      MOV      (SP)+, R0      ; GET MAPPED ADDRESS
201

```



\*\*FILE\*\*ID\*\*CF24C9

```

CCCCCCCC  FFFFFFFF  222222  AAAAAA  CCCCCCCC  999999
CCCCCCCC  FFFFFFFF  222222  AAAAAA  CCCCCCCC  999999
CC         FF        22        AA        CC        99
CC         FF        22        AA        CC        99
CC         FF        22        AA        CC        99
CC         FF        22        AA        CC        99
CC         FFFFFFFF  22        AA        CC        99999999
CC         FFFFFFFF  22        AA        CC        99999999
CC         FF        22        AAAAAAAAAA  CC        99
CC         FF        22        AAAAAAAAAA  CC        99
CC         FF        22        AA        AA  CC        99
CC         FF        22        AA        AA  CC        99
CCCCCCCC  FF        2222222222  AA        AA  CCCCCCCC  999999
CCCCCCCC  FF        2222222222  AA        AA  CCCCCCCC  999999

```

```

....
....
....
....

```

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LL          SSSSSS    TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT

```

```

393
394
395
396
397
398 001066
399
400
401 001066
402 001066
403 001066
404 001066
405 001066
406 001066
407 001066
408 001066
409 001066
410 001066
411 001066
412 001066
413 001066
414 001066
415 001066
416 001066
417 001066
418 001066
419 001066
420 001066
421 001066
422 001066
423 001066
424 001066
425 001066
426 001066
427 001066
428 001066
429 001066
430 001066
431 001066
432 001066
433 001066
434 001066
435 001066
436 001066
437 001066
438 001066
439 001066
440 001066
441 001066
442 001066
443 001066
444 001066
445 001066
446 001066
447 001066
448 001066
449 001066

```

:+ XPT'S LLC\$DF STATE TABLE FOR POOL.. ALLOCATION  
 :-

```

          ISTAT$ LLCST,LLCKW
          STATES$ BRADF
          TRANS$ %SLT$DF$%
          STATES$
          TRANS$ $RAD50
          STATES$
          TRANS$ <','>
          STATES$
          TRANS$ $RAD50
          STATES$
          TRANS$ <','>
          STATES$
          TRANS$ $RAD50
          STATES$
          TRANS$ <','>
          STATES$
          TRANS$ <','>,SLTDF2
          TRANS$ $LAMDA
          STATES$ SLTDF3
          TRANS$ %LF.MFL%,SLTDF1
          TRANS$ %LF.ENA%,SLTDF1
          TRANS$ %LF.SER%,SLTDF1
          TRANS$ $RAD50
          STATES$ SLTDF1
          TRANS$ '! ,SLTDF3
          TRANS$ <','>
          STATES$ SLTDF2
          TRANS$ $NUMBR
          STATES$
          TRANS$ <','>
          STATES$
          TRANS$ $NUMBR
          STATES$
          TRANS$ $EOS,$EXIT
          TRANS$ <','>
          STATES$
          TRANS$ <','>,SCTIM
          TRANS$ $STRNG
          STATES$
          TRANS$ $EOS,$EXIT
          TRANS$ <','>
          STATES$ SCTIM
          TRANS$ $NUMBR,$CHELO
          TRANS$ <','>,$CHEL
          STATES$ $CHELO
          TRANS$ $EOS,$EXIT
          TRANS$ <','>
          STATES$ $CHEL
          TRANS$ $NUMBR,BRADJ
          TRANS$ <','>

```

CF2DEC - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 <sup>D 15</sup> 00:14  
Table of contents

5- 45 MACRO DEFINITIONS  
7- 78 DEC\$DF ACTION ROUTINES

:

.TITLE CF2UMR - NTL INITIALIZATION UMR SETUP  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - INITIALIZATION UMR SETUP

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RX V1.0

```

252 000440 010061 000000G      MOV      R0,$X2MWS(R1)      ; STORE MAXIMUM WINDOW SIZE
253 000444      RETURN
254 000446      101$: MSG$R 2F      ; ILLEGAL MAXIMUM WINDOW SIZE VALUE
255
256      ; MAXIMUM BLOCK SIZE
257
258 000454 005767 000000G      X2.MBS: TST      .PNUMH      ; LEGAL MAXIMUM BLOCK SIZE VALUE?
259 000460 001010      BNE      101$      ; BR IF NO
260 000462 016700 000000G      MOV      .PNUMB,R0      ; GET MAXIMUM BLOCK SIZE
261 000466 016701 000000G      MOV      UCNT,R1      ; GET UNIT COUNT
262 000472 006301      ASL      R1      ; MAKE IT A WORD INDEX
263 000474 010061 000000G      MOV      R0,$X2MBS(R1)      ; STORE MAXIMUM BLOCK SIZE
264 000500      RETURN
265 000502      101$: MSG$R 2r      ; ILLEGAL MAXIMUM BLOCK SIZE
266
267      ; MAXIMUM RETRANSMIT COUNT
268
269 000510 005767 000000G      X2.MRT: TST      .PNUMH      ; LEGAL RETRANSMIT COUNT?
270 000514 001012      BNE      101$      ; BR IF NO
271 000516 105767 000001G      TSTB     .PNUMB+1      ;
272 000522 001007      BNE      101$      ; BR IF NO
273 000524 016701 000000G      MOV      UCNT,R1      ; GET UNIT COUNT
274 000530 006301      ASL      R1      ; MAKE IT A WORD INDEX
275 000532 016761 000000G 000000G      MOV      .PNUMB,$X2MRT(R1)      ; STORE MAXIMUM RETRANSMIT COUNT
276 000540      RETURN
277 000542      101$: MSG$R 2J      ; ILLEGAL MAXIMUM RETRANSMIT COUNT
278
279      ; RETRANSMIT TIMER
280
281 000550 005767 000000G      X2.RTR: TST      .PNUMH      ; LEGAL RETRANSMIT TIMER?
282 000554 001014      BNE      101$      ; BR IF NO
283 000556 016700 000000G      MOV      .PNUMB,R0      ; GET RETRANSMIT TIMER
284 000562 002411      BLT      101$      ; BR IF ILLEGAL VALUE
285 000564 022700 001130      CMP      #X2RTMX,R0      ; LEGAL RETRANSMIT TIMER VALUE?
286 000570 103406      BLO      101$      ; BR IF ILLEGAL VALUE
287 000572 016701 000000G      MOV      UCNT,R1      ; GET UNIT COUNT
288 000576 006301      ASL      R1      ; MAKE IT A WORD INDEX
289 000600 010061 000000G      MOV      R0,$X2RET(R1)      ; STORE RETRANSMIT TIMER
290 000604      RETURN
291 000606      101$: MSG$R 2K      ; ILLEGAL RETRANSMIT TIMER
292
293      ; HOLDBACK TIMER
294
295 000614 005767 000000G      X2.HLD: TST      .PNUMH      ; LEGAL HOLDBACK TIMER?
296 000620 001016      BNE      101$      ; BR IF NO
297 000622 016700 000000G      MOV      .PNUMB,R0      ; GET HOLDBACK TIMER VALUE
298 000626 002413      BLT      101$      ; BR IF ILLEGAL VALUE
299 000630 022700 001130      CMP      #X2HLMX,R0      ; LEGAL HOLDBACK TIMER VALUE?
300 000634 103410      BLO      101$      ; BR IF NO
301 000636 016701 000000G      MOV      UCNT,R1      ; GET UNIT COUNT
302 000642 006301      ASL      R1      ; MAKE IT A WORD INDEX
303 000644 010061 000000G      MOV      R0,$X2HLD(R1)      ; STORE HOLDBACK TIMER VALUE
304 000650 005267 000000G      INC      UCNT      ; UPDATE UNIT COUNT
305 000654      RETURN
306 000656      101$: MSG$R 2L      ; ILLEGAL HOLDBACK TIMER VALUE
307
308      000001      .END

```

LOCAL DATA

```

72          .SBTTL  LOCAL DATA
73
74 000000          .PSECT  DATA,D
75
76          : LOCAL DATA
77          :
78 000000          NEXT: .BLKW 1          : ADDRESS OF NEXT TIMER
79 000002          X3DEV: .BLKW 1          : RAD50 DEVICE NAME
80 000004          X3CTL: .BLKW 1          : CONTROLLER NUMBER
81 000006          X3UNT: .BLKW 1          : UNIT NUMBER
82 000010          TIMRCT: .BLKB 1          : COUNT OF NUMBER OF TIMER PARAMETERS SEEN
83          .EVEN

```

CF2A  
MACRO

CF2A

56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76

```
.SBTTL  MACRO DEFINITIONS

;****
; LIBRARY MACRO CALLS
;****

.MCALL  RNBDP$,EMSG$,RETC,NTLR$,ASL$,DHBDP$,SAVRG,RESRG
        DHBDP$                ; DEFINE DECNET HOME BLOCK OFFSETS
        RNBDP$                ; DEFINE REMOTE BLOCK OFFSETS

;***
; LOCAL MACRO DEFINITIONS
;***

.MACRO  SAVMAP
MOV     KSAR5,-(SP)           ; SAVE APR 5
.ENDM

.MACRO  RESMAP
MOV     (SP)+,KSAR5           ; RESTORE APR5
.ENDM
```

E 4

CREF 04.00

|          |        |        |        |        |        |        |        |        |        |        |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| ASL*     | #5-62  | 10-259 |        |        |        |        |        |        |        |        |
| CALL     | 10-271 | 10-272 | 10-279 | 10-287 | 10-305 | 10-311 |        |        |        |        |
| DHBD\$   | #5-62  | 5-64   |        |        |        |        |        |        |        |        |
| EMSG\$R  | #5-62  | 10-253 | 10-264 | 10-320 | 10-321 | 11-335 |        |        |        |        |
| NILER\$  | #5-62  | 7-99   | 7-100  | 7-101  | 7-102  | 7-103  |        |        |        |        |
| RESMAP   | #5-74  | 10-312 |        |        |        |        |        |        |        |        |
| RESRG    | #5-62  |        |        |        |        |        |        |        |        |        |
| RETIC    | #5-62  | 10-274 |        |        |        |        |        |        |        |        |
| RETURN   | 8-129  | 8-155  | 9-186  | 9-231  | 10-249 | 10-262 | 10-313 | 10-316 | 11-334 | 11-340 |
|          | 11-343 | 12-365 | 12-374 |        |        |        |        |        |        |        |
| RNBDF\$  | #5-62  | 5-65   |        |        |        |        |        |        |        |        |
| SAVMAP   | #5-70  | 10-276 |        |        |        |        |        |        |        |        |
| SAVRG    | #5-62  |        |        |        |        |        |        |        |        |        |
| \$WSTK\$ | 10-271 |        |        |        |        |        |        |        |        |        |

★★FI

CC  
CC

CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC

CC  
CC

[illegible]

F 4

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
84



\*\*FILE\*\*ID\*\*CF2AC3

```

CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCCCC 333333
CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCCCC 333333
CC        FF        22        AA        AA        CC        33        33
CC        FF        22        AA        AA        CC        33        33
CC        FF        22        AA        AA        CC        33        33
CC        FF        22        AA        AA        CC        33        33
CC        FFFFFFFF 22        AA        AA        CC        33        33
CC        FFFFFFFF 22        AA        AA        CC        33        33
CC        FF        22        AAAAAAAAAA CC        33        33
CC        FF        22        AAAAAAAAAA CC        33        33
CC        FF        22        AA        AA        CC        33        33
CC        FF        22        AA        AA        CC        33        33
CC        FF        22        AA        AA        CC        33        33
CCCCCCCC FF 2222222222 AA AA CCCCCCCC 333333
CCCCCCCC FF 2222222222 AA AA CCCCCCCC 333333

```

```

....
....
....
....

```

```

LL        SSSSSSSS TTTTTTTTTT
LL        SSSSSSSS TTTTTTTTTT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SSSSSS    TT
LL        SSSSSS    TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

CF2AC3  
SYME  
SYME  
BFEN  
BF TA  
B. EN  
B. NL  
B. ST  
B. ZE  
B. ZE  
CERR  
CFGE  
CFGE  
CFGS  
CFLI  
CRCR  
FMT8  
FM.8  
MSSA  
OROU  
ROU  
ROU  
RTSF  
R. A  
R. A  
R. A  
R. B  
R. C  
R. H  
R. M  
R. R  
SYNI  
SSA  
SAL  
SAN  
SBL  
SDI  
SDN  
SEO  
SER  
SER  
SER  
SER  
SER  
SER  
SER  
SER

## SYMBOL CROSS REFERENCE

CREF 04.00

## SYMBOL VALUE REFERENCES

|         |             |         |         |        |        |        |        |        |        |        |  |
|---------|-------------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--|
| BFEND   | = 000040 R  | #5-94   | 7-138   | 7-153  |        |        |        |        |        |        |  |
| BFTAB   | 000000 R    | #5-86   | 7-130   |        |        |        |        |        |        |        |  |
| B.END   | 000060 RG   | #7-153  |         |        |        |        |        |        |        |        |  |
| B.NUM   | 000014 RG   | #7-137  |         |        |        |        |        |        |        |        |  |
| B.STR   | 000000 RG   | #7-130  |         |        |        |        |        |        |        |        |  |
| B.ZEND  | 000054 RG   | #7-152  |         |        |        |        |        |        |        |        |  |
| B.ZERO  | 000010 RG   | #7-136  | 7-152   |        |        |        |        |        |        |        |  |
| CERR    | = ***** GX  | 6-104   | 6-105   | 6-106  | 6-107  | 6-108  | 6-109  | 6-110  | 6-111  | 6-112  |  |
|         |             | 6-113   | 6-114   | 6-115  | 6-116  |        |        |        |        |        |  |
| CFGBF   | = ***** GX  | 8-200   |         |        |        |        |        |        |        |        |  |
| CFGROU  | = 000000 RG | #12-319 |         |        |        |        |        |        |        |        |  |
| CFGSZ   | = ***** GX  | 8-199   |         |        |        |        |        |        |        |        |  |
| CFLIN   | = ***** GX  | 6-104   | 6-105   |        |        |        |        |        |        |        |  |
| CRC\$OV | = ***** GX  | 7-162   | 7-169   |        |        |        |        |        |        |        |  |
| FMT8    | = ***** GX  | 6-104   | 6-105   | 6-106  | 6-107  | 6-108  | 6-109  | 6-110  | 6-111  | 6-112  |  |
|         |             | 6-113   | 6-114   | 6-115  | 6-116  |        |        |        |        |        |  |
| FM.8    | = 000000    | #6-104  | #6-105  | #6-106 | #6-107 | #6-108 | #6-109 | #6-110 | #6-111 | #6-112 |  |
|         |             | #6-113  | #6-114  | #6-115 | #6-116 |        |        |        |        |        |  |
| M\$MGE  | = 000000    | 7-177   |         |        |        |        |        |        |        |        |  |
| QROU    | 000224 RG   | #8-196  |         |        |        |        |        |        |        |        |  |
| ROUDF   | 000000 R    | 8-196   |         |        |        |        |        |        |        |        |  |
| ROUKW   | 000000 RG   | 8-198   |         |        |        |        |        |        |        |        |  |
| RTSPC   | = ***** GX  | 6-104   | #12-319 |        |        |        |        |        |        |        |  |
|         |             | 6-113   | 6-105   | 6-106  | 6-107  | 6-108  | 6-109  | 6-110  | 6-111  | 6-112  |  |
|         |             | 6-114   | 6-114   | 6-115  | 6-116  |        |        |        |        |        |  |
| R.ADDR  | 000276 RG   | #9-218  |         |        |        |        |        |        |        |        |  |
| R.AMXC  | 000544 RG   | #10-288 |         |        |        |        |        |        |        |        |  |
| R.AMXH  | 000516 RG   | #10-279 |         |        |        |        |        |        |        |        |  |
| R.BEA   | 000452 RG   | #10-265 |         |        |        |        |        |        |        |        |  |
| R.BTM   | 000572 RG   | #11-298 |         |        |        |        |        |        |        |        |  |
| R.COST  | 000306 RG   | #9-222  |         |        |        |        |        |        |        |        |  |
| R.HOP   | 000334 RG   | #9-229  |         |        |        |        |        |        |        |        |  |
| R.MAXA  | 000406 RG   | #9-245  |         |        |        |        |        |        |        |        |  |
| R.RTM   | 000362 RG   | #9-237  |         |        |        |        |        |        |        |        |  |
| SYNERR  | = ***** GX  | *8-201  | *8-204  |        |        |        |        |        |        |        |  |
| S\$BAS  | = *****     | 6-104   | 6-104   | 6-105  | 6-105  | 6-106  | 6-106  | 6-107  | 6-107  | 6-108  |  |
|         |             | 6-108   | 6-109   | 6-109  | 6-110  | 6-110  | 6-111  | 6-111  | 6-112  | 6-112  |  |
|         |             | 6-113   | 6-113   | 6-114  | 6-114  | 6-115  | 6-115  | 6-116  | 6-116  |        |  |
| \$ALPHA | = 000022    | #12-319 |         |        |        |        |        |        |        |        |  |
| \$ANY   | = 000020    | #12-319 |         |        |        |        |        |        |        |        |  |
| \$BLANK | = 000006    | #12-319 |         |        |        |        |        |        |        |        |  |
| \$DIGIT | = 000024    | #12-319 |         |        |        |        |        |        |        |        |  |
| \$DNUMB | = 000014    | #12-319 |         |        |        |        |        |        |        |        |  |
| \$EQS   | = 000012    | #12-319 |         |        |        |        |        |        |        |        |  |
| \$ERRXR | 000040 R    | #6-104  | 7-147   |        |        |        |        |        |        |        |  |
| \$ERRXS | 000074 R    | #6-105  | 7-191   |        |        |        |        |        |        |        |  |
| \$ERRYI | 000632 R    | #6-115  | 11-309  |        |        |        |        |        |        |        |  |
| \$ERRYJ | 000700 R    | #6-116  |         |        |        |        |        |        |        |        |  |
| \$ERRYK | 000126 R    | #6-106  | 9-227   |        |        |        |        |        |        |        |  |
| \$ERRYL | 000162 R    | #6-107  | 9-234   |        |        |        |        |        |        |        |  |
| \$ERRYM | 000214 R    | #6-108  | 9-257   |        |        |        |        |        |        |        |  |
| \$ERRYN | 000260 R    | #6-109  |         |        |        |        |        |        |        |        |  |
| \$ERRYO | 000342 R    | #6-110  | 10-277  |        |        |        |        |        |        |        |  |

CF2AC4      CREATED BY    MACRO    ON 29-JUN-85 AT 00:11      PAGE 1      E 7

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL   | VALUE      | REFERENCES   |
|----------|------------|--|
| CERR     | = ***** GX | 7-104    7-105    7-106    7-107                                       |
| CLASS    | 000000 R   | #6-90    *8-128    9-193    9-209                                      |
| EVENT    | 000004 R   | #6-92    8-130    8-149    9-195    9-196    9-197    9-198            |
| E.CON    | 000130 RG  | #9-173   |
| E.FIL    | 000140 RG  | #9-176   |
| E.MON    | 000150 RG  | #9-179   |
| E.MSK    | 000060 RG  | #8-147   |
| E.NUM    | 000000 RG  | #8-123   |
| FF.ADD   | = 000020   | #5-63    5-63  |
| FF.CIR   | = 000040   | #5-63    5-63  |
| FF.CON   | = 000001   | #5-63    9-173   |
| FF.FIL   | = 000002   | #5-63    9-176   |
| FF.HST   | = 040000   | #5-63  |
| FF.LIN   | = 000010   | #5-63    5-63  |
| FF.MOD   | = 000100   | #5-63    5-63  |
| FF.MON   | = 000004   | #5-63    9-179   |
| FF.MSK   | = 000077   | #5-63  |
| FF.PRT   | = 000200   | #5-63    5-63  |
| FF.QL    | = 000370   | #5-63  |
| FF.REM   | = 100000   | #5-63  |
| FILHD    | = ***** GX | 9-200  |
| FLAG     | 000002 R   | #6-91    *8-129    *9-173    *9-176    *9-179    9-199                 |
| FMT8     | = ***** GX | 7-104    7-105    7-106    7-107                                       |
| FM.8     | = 000000   | #7-104    #7-105    #7-106    #7-107                                   |
| F.ADD    | 000016     | #5-63  |
| F.CEV    | 000030     | #5-63  |
| F.CIR    | 000024     | #5-63  |
| F.CLS    | 000002     | #5-63    *9-193    *9-194    9-209                                     |
| F.EVT    | 000004     | #5-63    *9-195    *9-196    *9-197    *9-198                          |
| F.FLG    | 000014     | #5-63    *9-199  |
| F.LEN    | 000016     | #5-63    9-181    9-194  |
| F.LIN    | 000020     | #5-63  |
| F.LNK    | 000000     | #5-63  |
| F.MCD    | 000022     | #5-63  |
| F.REM    | 000026     | #5-63  |
| F.SEV    | 000004     | #5-63  |
| KSAR5    | = ***** GX | 9-188    *9-214  |
| NS\$VCT  | = *****    | 9-186  |
| RTSPC    | = ***** GX | 7-104    7-105    7-106    7-107                                       |
| STRADD   | 000014 R   | #6-94    *8-131    8-149    8-151    *8-152    7-106    7-107    7-107 |
| \$B\$BAS | = *****    | 7-104    7-105    7-105    7-105                                       |
| \$CEACX  | = ***** GX | 9-191    9-207   |
| \$ERRE1  | 000016 R   | #7-104    8-139  |
| \$ERRE2  | 000052 R   | #7-105    8-158  |
| \$ERRE3  | 000120 R   | #7-106    9-222  |
| \$ERRE4  | 000206 R   | #7-107    8-159  |
| \$HEADR  | = ***** GX | 9-186  |
| \$LINKX  | = ***** GX | 9-213  |
| \$XALOC  | = ***** GX | 9-184  |
| .PNUMB   | = ***** GX | 8-125    8-151   |
| .PNUMH   | = ***** GX | 8-123    8-147   |

CF2AC5 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:11 Page 11-1  
CRFNBK - CREATE FILENAME DESCRIPTOR BLOCK

316 000500  
317  
318

30\$: RETURN

CF2AC5  
MACRO

```
53 .SBTTL MACRO DEFINITIONS
54
55 ;***
56 ; LIBRARY MACROS
57 ;***
58 .MCALL RESMAP,SAVMAP,EMSG$R,SLTDF$,NTLR$,ASL$,DHBDF$,SAVRG,RESRG
59
60 ;***
61 ; LIBRARY SYMBOLS
62 ;***
63 000000 SLTDF$ ; DEFINE SLT OFFSETS
64 000000 DHBDF$ ; DEFINE DECNET HOME BLOCK OFFSETS
65
66
67 .MACRO SAVMAP
68 MOV @KSAR5,-(SP) ; SAVE APR5 MAPPING
69 .ENDM
70
71 .MACRO RESMAP
72 MOV (SP)+,@KSAR5 ; RESTORE APR5 MAPPING
73 .ENDM
74
75
76
```

**E 10**

CREF 04.00

[illegible]CF2  
Syn

ALC  
AT\$  
AT\$  
AT\$  
AT\$  
AT\$  
AT\$  
ASL  
ASN  
AST  
AST  
AST  
AS\$  
AS\$  
AS\$  
CL\$  
CL\$  
CL  
CL  
CR  
CV\$  
CV\$  
CV\$  
CV\$  
C\$  
C\$  
DE  
D\$  
D\$  
D\$  
D\$  
EC  
ER  
ER  
ER  
ER  
ER  
ER  
ER  
ER  
ER  
ER

**F 10**

**F 10**

|                  |                  |                |                |                  |
|------------------|------------------|----------------|----------------|------------------|
| ALOCB = ***** GX | ESNBR 000014     | LO.PRI 000010  | L\$11R= 000000 | MD.ILS= 000040   |
| ATSACL= 000100   | ESNBS 000020     | LO.SCR 000014  | L.CSTA 000037  | MD.IM = 000020   |
| ATSADN= 000040   | ESNCR 000034     | LO.STA 000015  | L.CTR 000074   | MF.ACK= 000004   |
| ATSAUP= 000020   | ESNCS 000036     | LO.VCT 000012  | L.DCR 000100   | MF.CTL= 000010   |
| ATSCYC= 000004   | ESNIC 000044     | LR.AST 000026  | L.FLAG 000014  | MF.DAT= 000000   |
| ATSLV1= 000002   | ESNLEN 000050    | LR.CIR 000002  | L.ILSQ 000052  | M\$CRB= 000124   |
| ATSLV2= 000001   | ESNLLA 000012    | LR.CTL 000004  | L.ILTT 000066  | M\$CRX= 000000   |
| ATSNEX= 000200   | ESNLNK 000000    | LR.EFN 000004  | L.LDA 000032   | M\$FCS= 000000   |
| ATSUP = 000010   | ESNML 000040     | LR.LIN 000002  | L.LIA 000034   | M\$MGE= 000000   |
| ASCIR 000003     | ESNMR 000024     | LR.PRM 000030  | L.LLA 000002   | M\$NET= 000000   |
| ASLEN 000004     | ESNMS 000030     | LR.PRO 000002  | L.LNG 000124   | M\$OVR= 000000   |
| ASNID 000000     | ENNOD 000002     | LR.STS 000000  | L.LNO 000026   | NC.FM0= 000000   |
| ASTM 000003      | ESNRT 000042     | LR.TCB 000002  | L.LPT 000065   | NC.FM1= 000001   |
| ASTMI 000002     | ESNRT 000005     | LR.UNT 000005  | L.LSA 000030   | NC.FM2= 000002   |
| ASTSZ 000000     | ESNSEG 000010    | LS.AST= 100000 | L.LSFD 000046  | NSDDON= 000000   |
| ASTYP 000002     | ESNTIM 000046    | LS.CEX= 000001 | L.LSFI 000044  | NSSDI= 000002    |
| AS\$CHK= 000000  | ESNUSE 000004    | LS.CIR= 000010 | L.LTT 000062   | NS\$WDC= 000004  |
| AS\$CPS= 000000  | ESSTR 000006     | LS.CXD= 010000 | L.MASQ 000070  | NS\$ACC= 000001  |
| AS\$PRI= 000000  | ES\$XPR= 000000  | LS.CX5= 000400 | L.MAST 000073  | NS\$BUF= 000001  |
| AS\$TRP= 000000  | FE.EXT= ***** GX | LS.DLM= 004000 | L.MASZ 000072  | NS\$LDV= 000001  |
| CL\$MFL= 000010  | FE.PLA= ***** GX | LS.DLS= 100000 | L.NIN 000020   | NS\$MCP= 000001  |
| CL\$SFL= 000004  | FMASK = ***** GX | LS.ECH= 001000 | L.NXN 000016   | NS\$MLL= 000001  |
| CL\$TYP= 000001  | F\$SLVL= 000001  | LS.FCC= 000004 | L.NXTH 000010  | NS\$MOV= 000010  |
| CL.MU1= 000001   | G\$STPP= 000000  | LS.FCD= 000001 | L.OPDL 000102  | NS\$NCT= 000001  |
| CL.MU2= 000002   | G\$STSS= 000000  | LS.FCI= 000002 | L.REM 000006   | NS\$PEM= 000001  |
| CL.RES= 177774   | G\$STTK= 000000  | LS.FDX= 004000 | L.RFC 000050   | PARHD = ***** GX |
| CRSUB 000736R    | G\$STWRD= 000000 | LS.HDX= 010000 | L.RLA 000004   | PLGTH = ***** GX |
| CV\$MSK= 000003  | IN.DAT= 000400   | LS.ILS= 100000 | L.RNO 000022   | PS.COM= ***** GX |
| CV\$31 = 000001  | IN.ILS= 000001   | LS.LIN= 000003 | L.RTO 000060   | PS.NSF= ***** GX |
| CV\$32 = 000000  | IS\$RAR= 000000  | LS.LMC= 000007 | L.RTYD 000055  | PS.SYS= ***** GX |
| CV\$40 = 000002  | IS\$RDN= 000000  | LS.MAK= 000020 | L.RTYI 000057  | PS\$P45= 000000  |
| C\$CKP= 000000   | K\$SCNT= 177546  | LS.MNK= 000040 | L.SEC 000064   | PS\$WRD= 000000  |
| C\$ORE= 000400   | K\$SCSR= 177546  | LS.NTJ= 004000 | L.SEGZ 000076  | P.ALL 000002RG   |
| C\$RSH= 177564   | K\$SLDC= 000000  | LS.ON = 000011 | L.STA 000000   | P.ATT = ***** GX |
| DEACB = ***** GX | K\$STPS= 000074  | LS.OPT= 000400 | L.TIC 000042   | P.BLSK= ***** GX |
| D\$BUG= 177514   | K6TMP 000000R    | LS.PRO= 000002 | L.TIC 000043   | P.END 000016RG   |
| D\$ISK= 000000   | LA.ACK= 100000   | LS.PWF= 040000 | L.TIPD 000013  | P.MAIN= ***** GX |
| D\$SL11= 000001  | LA.CRS= 020000   | LS.RES= 000360 | L.TIPI 000012  | P.NAM = ***** GX |
| D\$SYNC= 000000  | LA.MSK= 170000   | LS.RSV= 000300 | L.TMRD 000054  | P.PRO = ***** GX |
| D\$SYNM= 000000  | LA.NAK= 110000   | LS.TOP= 020000 | L.TMRI 000056  | P.REL = ***** GX |
| ECLLLN= ***** GX | LA.NMS= 010000   | LS.UNF= 020000 | L.TYP 000001   | P.STAT= ***** GX |
| ECLNOD= ***** GX | LA.RES= 040000   | LS.X12= 000012 | L.USA 000024   | P.SUB = ***** GX |
| ER\$ABM= 000010  | LA.WND= 004000   | LS.T1D= 004000 | L.USTA 000036  | Q\$OPT= 000010   |
| ER\$ABO= 000046  | LD\$LP = 000000  | LT.CCA= 000020 | L.VER 000015   | R\$DER= 000000   |
| ER\$ABT= 000011  | LF.DRD= 000004   | LT.DIR= 000010 | L.WIND 000040  | R\$K11= 000001   |
| ER\$ACC= 000042  | LF.FRC= 000001   | LT.LCL= 000001 | MA.CI = 000040 | R\$SND= 000000   |
| ER\$CDI= 000052  | LF.HFO= 000010   | LT.LPL= 000002 | MA.DA = 000000 | R\$11M= 000000   |
| ER\$COM= 000047  | LF.HMF= 000040   | LT.NOT= 000040 | MA.IL = 000020 | ST\$CC = 000004  |
| ER\$FMT= 000005  | LF.HSF= 000020   | LT.RSU= 000200 | MC.CC = 000040 | ST\$CIR= 000006  |
| ER\$MLB= 000006  | LF.IRD= 000002   | LT.SLI= 000004 | MC.CI = 000020 | ST\$CIS= 000002  |
| ER\$NMF= 000012  | LF.MMF= 000200   | LT.TDA= 000100 | MC.DC = 000100 | ST\$DAT= 000010  |
| ER\$NOD= 000002  | LF.MSF= 000100   | LX.CEX= 000004 | MC.DI = 000060 | ST\$DIP= 000012  |
| ER\$NSL= 000013  | LO.COM 000006    | LX.LIN= 000006 | MC.NO = 000000 | ST\$PND= 000014  |
| ER\$NSR= 000003  | LO.CSR 000006    | LX.PRO= 000005 | MC.RC = 000140 | ST\$WRG= 000000  |
| ER\$RES= 000001  | LO.CTL 000014    | L\$ASG= 000000 | MD.BM = 000040 | S\$YSZ= 007600   |
| ER\$STA= 000057  | LO.LIN 000004    | L\$DRV= 000000 | MD.EM = 000100 | T\$FLAG 000044   |
| ER\$UOB= 000004  | LO.PAR 000010    | L\$P11= 000001 |                | T\$LIF 000013    |

```

202 000236 010003      MOV      R0,R3          ; SAVE MAPPED ADDRESS
203 000240 012701      MOV      #S LENG,R1       ; SET BLOCK SIZE
204 000244 105020      CLR      (R0)+          ; ZERO OUT BLOCK
205 000246 005301      DEC      R1              ; COUNT ONE LESS BYTE
206 000250 001375      BNE      15$             ; IF MORE - BRANCH
207 000252 010300      MOV      R3,R0          ; RESTORE MAPPED ADDRESS
208
209 000254 056760 000000G 000002      BIS      SERFLG,S.FLAG(R0) ; CLEAR FLAG WORD
210 000262 016760 000000G 000004      MOV      SERNOD,S.ADD(R0) ; GET NODE ADDRESS
211 000270 016760 000000G 000006      MOV      SERHST,S.HST(R0) ; GET HOST ADDRESS
212
213 000276 016760 000000G 000010      MOV      SERADD,S.HAD(R0) ; GET HARDWARE ADDRESS
214 000304 016760 000002G 000012      MOV      SERADD+2,S.HAD+2(R0) ; ...
215 000312 016760 000004G 000014      MOV      SERADD+4,S.HAD+4(R0) ; ...
216 000320 016760 000000G 000016      MOV      SERSLT,S.CIR(R0) ; GET SLN/STA PAIR
217
218 000326 016760 000000G 000020      MOV      SERDEV,S.DEV(R0) ; GET DEVICE NAME RAD50
219 000334 016760 000000G 000022      MOV      SERPAS,S.PSS(R0) ; GET SERVICE PASSWORD
220 000342 016760 000002G 000024      MOV      SERPAS+2,S.PSS+2(R0) ; ...
221
222 000350 016760 000004G 000026      MOV      SERPAS+4,S.PSS+4(R0) ; ...
223 000356 016760 000006G 000030      MOV      SERPAS+6,S.PSS+6(R0) ; ...
224 000364 016760 000000G 000032      MOV      SERDMP,S.DPA(R0) ; GET DUMP ADDRESS
225 000372 016760 000002G 000034      MOV      SERDMP+2,S.DPA+2(R0) ; ...
226 000400 016760 000000G 000036      MOV      SERDCT,S.DPC(R0) ; GET DUMP COUNT
227
228 000406 017700 000000G      MOV      @DECPT,R0      ; STORE DECNET HOME BLOCK ADDRESS
229 000412 062700 000032      ADD      #D$SER,R0      ; POINT TO SERVICE BLOCK LISTHEAD
230 000416 010003      MOV      R0,R3          ; SAVE LISTHEAD
231
232 000420 010301      MOV      R3,R1          ; STORE CURRENT UNMAPPED ADDRESS
233 000422 011000      MOV      (R0),R0      ; GET NEXT SERVICE BLOCK
234 000424 001406      BEQ      25$             ; IF END OF LIST - BRANCH
235
236 000426 010003      MOV      R0,R3          ; SAVE UNMAPPED ADDRESS OF NEW SERVICE BLOCK
237 000430 010046      MOV      R0,-(SP)        ; SET UP UNMAPPED ADDRESS FOR ACCESS
238 000432      CALL      $CEACX          ; MAP TO BLOCK
239 000436 012600      MOV      (SP)+,R0      ; GET MAPPED ADDRESS
240 000440 000767      BR       20$             ; CONTINUE
241
242 000442 010246      MOV      R2,-(SP)        ; SET UP UNMAPPED ADDRESS FOR $LINKX
243 000444 010146      MOV      R1,-(SP)        ; ...
244 000446      CALL      $LINKX          ; LINK NEW SERVICE BLOCK INTO LIST
245 000452      RESMAP          ; RESTORE MAPPING
246
247 000456      30$:      RETURN
248
249
250

```



CF2AC9 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:13 **E 13**  
Table of contents

5- 49 MACRO DEFINITIONS  
7- 108 ACTION ROUTINES FOR PAR\$D+  
9- 322 SER\$DF ACTION ROUTINES

CF2  
SER

CF2AC9 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:13 Page 10-1  
SER\$DF ACTION ROUTINES

450 001066  
451 001066  
452 001066  
453 001066  
454 001066

STATES\$ BRADJ  
TRANS\$ <'>,BRADJ1  
TRANS\$ \$EOS,\$EXIT  
STATES\$ BRADJ1  
TRANS\$ \$NUMBR,\$EXIT,BRDJ

.TITLE CF2DEC - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

.. COPYRIGHT (C) 1984, 1985  
.. DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

.. THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
.. ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
.. INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
.. COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
.. OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
.. TRANSFERRED.

.. THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
.. AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
.. CORPORATION.

.. DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
.. SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

.. MODULE DESCRIPTION:

.. NTL - CONFIG FILE ACTION ROUTINES

.. DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

.. IDENT HISTORY:

.. 4.00 22-JUN-84  
.. SUPPORT DEC\$DF MACRO

.. 5.00 22-JUL-85  
.. DECnet-11M/S V4.2  
.. DECnet-11M-Plus V3.0  
.. DECnet-Micro/RX V1.0

```

54          ;****
55          ; LIBRARY MACRO CALLS
56          ;****
57
58          .MCALL  MSG$R
59
60          .IF NDF I$$AS
61
62          .MCALL  NHWDF$
63
64          NHWDF$          ; DEFINE HARDWARE OFFSETS
65
66          .ENDC  ; NDF I$$AS
67
68
69          ;+
70          $UMRAL - ALLOCATE AND LOAD UMR'S FROM NETWORK POOL
71
72          : INPUTS:
73          R0 = BIAS OF CPU PARTITION (ONLY ON MULTI-PROCESSOR 11M+ SYSTEMS)
74          R1 = POOL PCB ADDRESS
75          .ALLOC = NUMBER OF BLOCKS OF EXTENDED NETWORK POOL
76          .NOMAP = NUMBER OF BLOCKS OF NON-UMR MAPPED NETWORK POOL
77          .UMRFL = UMR ENABLED/DISABLED FLAG
78
79          : OUTPUTS:
80          R1 = UNCHANGED
81          R2 = UMR BLOCK ADDRESS OR 0
82          .UMRBA = UMR BLOCK ADDRESS
83
84          ;-
85          .IF DF R$$11M & M$$MGE
86          $UMRAL::CLR R2          ; ASSUME FAILURE
87          BIT #FE,EXT,@FMASK      ; IS THIS REALLY AN 11/70?
88          BEQ 10$                ; IF EQ, NO
89          TST UMRFL              ; HAVE UMR'S BEEN DISABLED?
90          BNE 10$                ; IF NE, YES
91          SWSTK$ 20$             ; * ENTER KERNEL MODE
92          MOV #M.LGTH,R1         ; * GET UMR BLOCK SIZE
93          CALL @ALOCB            ; * TRY TO ALLOCATE
94          BCS 10$                ; * IF CS, FAILURE
95          MOV R0,6(SP)           ; * RETURN ADDRESS IN USER MODE R2
96          RETURN                 ; * BACK TO USER MODE
97          TST R2                 ; UMR BLOCK ADDRESS ?
98          BEQ 101$              ; IF ZERO, ALLOCATION FAILURE
99          JSR R5,$$SAVRG         ; SAVE R3, R4, AND R5 (CO-ROUTINE)
100         MOV P.BLK$ (R1),R3     ; GET POOL SIZE IN BLOCKS
101         SUB .ALOC,R3           ; LESS EXTENDED POOL ALLOCATIONS
102         SUB .NOMAP,R3          ; LESS NTL BLOCKS, ETC.
103         ADD #177,R3            ; ROUND UP TO NEXT MULTIPLE OF 4K WORDS
104         BIC #177,R3            ; ...
105         ASH #-5,R3             ; GIVING NUMBER OF UMR'S NEEDED * 4
106         MOV R3,M.UMRN(R2)      ; STASH COUNT IN UMR BLOCK
107         MOV P.REL(R1),R5       ; GET POOL STARTING BLOCK NUMBER
108         ADD .NOMAP,R5          ; LESS NTL BLOCKS, ETC.
109         CLR R4                 ; CONVERT TO A REAL MEMORY ADDRESS
110         ASHC #6,R4             ;
111         MOV R5,M.BFVL(R2)      ; STASH it IN UMR BLOCK

```

|                   |                |                  |                 |                     |
|-------------------|----------------|------------------|-----------------|---------------------|
| ASSCHK= 000000    | LF.SER= 000040 | MSSNET= 000000   | X2DEV 000000R   | Z.LLN 000006        |
| ASSCPS= 000000    | LF.TIM= 000010 | MSSOVR= 000000   | X2HLMX= 001130  | Z.MAP 000020        |
| ASSPRI= 000000    | LF.UNL= 020000 | NSSACC= 000001   | X2PDF 000000R   | 002 Z.NAM 000004    |
| ASSTRP= 000000    | LF.X2P= 000000 | NSSBUF= 000001   | X2PKW 000000RG  | 003 Z.PCB 000012    |
| CFERR = ***** GX  | LN.CLO= 000000 | NSSLDV= 000001   | X2PRM 000036R   | 002 Z.SCH 000007    |
| CFGFBF = ***** GX | LN.DUM= 000005 | NSSMCP= 000001   | X2PST 000000RG  | 002 Z.ALPHA= 000022 |
| CFGSZ = ***** GX  | LN.LOA= 000004 | NSSMLL= 000001   | X2RTMX= 001130  | SANY = 000020       |
| CFLIN = ***** GX  | LN.LOO= 000003 | NSSMOV= 000010   | X2UNT 000003R   | SBLANK= 000006      |
| CHKDEV 000316R    | LN.OAU= 000003 | NSSNCT= 000001   | X2WNMN= 000001  | \$CAT5 = ***** GX   |
| CSSCKP= 000000    | LN.OFF= 000001 | NSSPEM= 000001   | X2WNMX= 000007  | \$DIGIT= 000024     |
| CSSORE= 000400    | LN.ON = 000000 | PDVTA = ***** GX | X2.CTL 000244R  | \$DNUMB= 000014     |
| CSSRSH= 177564    | LN.ODP= 000004 | PSSP45= 000000   | X2.DEV 000216R  | \$EOS = 000012      |
| DSSBUG= 177514    | LN.OPE= 000001 | PSSWRD= 000000   | X2.HLD 000614R  | \$ERR1T= ***** GX   |
| DSSISK= 000000    | LN.REF= 000002 | QSSOPT= 000010   | X2.MBS 000454R  | \$ERR2C= ***** GX   |
| DSSL11= 000001    | LN.SER= 000002 | RTSPC = ***** GX | X2.MRT 000510R  | \$ERR2F= ***** GX   |
| DSSYNC= 000000    | LN.STA= 000017 | RSSDER= 000000   | X2.MWS 000404R  | \$ERR2I= ***** GX   |
| DSSYNM= 000000    | LN.SUB= 000360 | RSSK11= 000001   | X2.RTR 000550R  | \$ERR2J 000004R     |
| ESSXPR= 000000    | LN.TRI= 000006 | RSSNDK= 000000   | X2.UNT 000274R  | \$ERR2K 000046R     |
| FMT8B = ***** GX  | LSSASG= 000000 | RSS11M= 000000   | ZF.COUL= 001000 | \$ERR2L 000106R     |
| FM.8B = 000000    | LSSDRV= 000000 | SF.ACT= 000200   | ZF.DDM= 000001  | \$EX1T = 000000     |
| FSSLVL= 000001    | LSSP11= 000001 | SF.ENA= 000100   | ZF.DIA= 004000  | \$FAIL = 177777     |
| GSSTPP= 000000    | LSS11R= 000000 | SF.LPB= 000004   | ZF.DLC= 000002  | \$LAMDA= 000000     |
| GSSSTS= 000000    | L.COST 000015  | SF.MFL= 000040   | ZF.DVP= 100000  | \$NUMBR= 000002     |
| GSSTTK= 000000    | L.CTL 000012   | SF.PAC= 000020   | ZF.INJ= 040000  | \$QX2P 000144RG     |
| GSSWRD= 000000    | L.CVA 177776   | SF.REA= 000010   | ZF.KMX= 000020  | \$RAD50= 000016     |
| ISSRAR= 000000    | L.DDM 000002   | SF.SER= 000001   | ZF.LLC= 000004  | \$SLTA = ***** GX   |
| ISSRDN= 000000    | L.DDS 000004   | SF.SVC= 000002   | ZF.LMC= 000100  | \$STRNG= 000004     |
| KSSCNT= 177546    | L.DLC 000003   | SF.UNL= 000040   | ZF.MAN= 020000  | \$SUBXP= 000010     |
| KSSCSR= 177546    | L.DLM 000006   | SYNERR= ***** GX | ZF.MFL= 000010  | \$X2HLD= ***** GX   |
| KSSLDC= 000000    | L.DLS 000010   | SSSWRG= 000000   | ZF.MTM= 000400  | \$X2MBS= ***** GX   |
| KSSTPS= 000074    | L.FLG 000000   | SSSYSZ= 007600   | ZF.MUX= 000040  | \$X2MRT= ***** GX   |
| LD\$LP = 000000   | L.KRBA 000016  | S.COST 000001    | ZF.PSE= 002000  | \$X2MWS= ***** GX   |
| LF.ACT= 100000    | L.LEN = 000022 | S.FLG 000000     | ZF.SLI= 010000  | \$X2RET= ***** GX   |
| LF.BRO= 000400    | L.MPF 000022   | S.LEN 000004     | ZF.TIM= 000200  | \$SFLG= 177777      |
| LF.BWT= 000007    | L.NMST 000020  | S.NMST 000002    | ZF.X3P= 000000  | \$SKEY= 000000      |
| LF.ENA= 002000    | L.NSTA 000014  | S.OWNR 000003    | ZS.ASN= 100000  | \$SSTA= 000000      |
| LF.LPB= 001000    | L.OWNR 000021  | TSSKMG= 000000   | ZS.BSY= 140000  | \$STMP= 000000R     |
| LF.MDC= 000100    | L.UNT 000013   | TSSMIN= 000000   | Z.AVL 000014    | .PNUMB= ***** GX    |
| LF.MFI= 004000    | MSSCRB= 000124 | UCNT = ***** GX  | Z.DAT 000016    | .PNUMH= ***** GX    |
| LF.MTP= 000020    | MSSCRX= 000000 | VSSCTR= 001000   | Z.DSP 000000    | .PSTCN= ***** GX    |
| LF.PAC= 000200    | MSSFCS= 000000 | XSSDBT= 000000   | Z.FLG 000010    | .PSTPT= ***** GX    |
| LF.RDY= 040000    | MSSMGE= 000000 | X2CTL 000002R    | Z.LEN = 000016  | .TPARS= ***** GX    |
| LF.REA= 010000    |                |                  |                 |                     |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
 000664 001 (RW,I,LCL,REL,CON)  
 \$STATE 000102 002 (RW,D,LCL,REL,CON)  
 \$KTAB 000002 003 (RW,D,LCL,REL,CON)  
 \$KSTR 000007 004 (RW,D,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 15400 Words ( 61 Pages)

```
85 .SBTT! ERROR MESSAGES
86
87
88
89
90
91 .ENABL LC
92 NTLERS$ ,2B,8B,CERR,RTSPC,CFLIN,<default block size>
93 000046 NTLERS$ ,2D,8,CERR,RTSPC,CFLIN,<Default block size exceeds maximum>
94 000122 NTLERS$ ,2E,8B,CERR,RTSPC,CFLIN,<default window size>
95 000136 NTLERS$ ,2G,8,CERR,RTSPC,CFLIN,<Default window size exceeds maximum>
96 000232 NTLERS$ ,2H,8B,CERR,RTSPC,CFLIN,<timer value>
97 .DSABL LC
98 .PSECT
```

```
LOCAL DATA
78          .SBTTL  LOCAL DATA
79          ;****
80          ; LOCAL DATA
81          ;****
82
83 000000          .PSECT  DATA,D
84
85          ; REMOTE NODE NAME
86          ;
87 000000          NAME:  .BLKB  6          ; REMOTE NAME
88 000006          RADDR: .BLKW  1          ; REMOTE ADDRESS
89 000010 000000          NAMED: .WORD  0          ; NAMFD DIRECTORY FLAG
```

FILE\*\*ID\*\*CF2AC2

F 4

```
CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCC 222222
CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCC 222222
CC FF 22 22 AA AA CC 22 22
CC FF 22 22 AA AA CC 22 22
CC FF 22 22 AA AA CC 22 22
CC FFFFFFFF 22 AA AA CC 22
CC FFFFFFFF 22 AA AA CC 22
CC FF 22 AA AA CC 22
CC FF 22 AAAAAAAAAA CC 22
CC FF 22 AAAAAAAAAA CC 22
CC FF 22 AA AA CC 22
CC FF 22 AA AA CC 22
CCCCCCCC FF 2222222222 AA AA CCCCCC 2222222222
CCCCCCCC FF 2222222222 AA AA CCCCCC 2222222222
```

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```



CF2AC3 - NTLCF2 ACTION ROUTINES MACRO V05.03b Saturday 29-Jun-85 00:10 <sup>6 5</sup>  
 Table of contents

|     |     |                                   |
|-----|-----|-----------------------------------|
| 5-  | 53  | MACRO DEFINITIONS                 |
| 5-  | 64  | LOCAL DATA                        |
| 6-  | 96  | ERROR MESSAGES                    |
| 7-  | 123 | TPARS ACTION ROUTINES FOR BUF\$DF |
| 8-  | 193 | TPARS ACTION ROUTINES FOR ROU\$DF |
| 12- | 314 | TPARS STATE TABLES - ROU\$DF      |

CF2AC3

SYME

SYME

\$ERR

\$ERR

\$ERR

\$ERR

\$ERR

\$EXI

\$FAI

\$GPP

\$LAN

\$MUL

\$NUN

\$RAC

\$RAC

\$RAC

\$STI

\$SUB

\$\$\$\$

\$\$\$\$

\$\$\$\$

.AM

.AM

.BFI

.CCI

.CCI

.MA

.MA

.MXI

.NA

.NBI

.NN

.NT

.PN

.PN

.RD

.RD

.RD

.RO

.RO

.SD

.SD

.TP

CF2AC3      CREATED BY    MACRO    ON 29-JUN-85 AT 00:10      PAGE 2      F 6

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL    | VALUE      | REFERENCES  |
|-----------|------------|---|
| \$ERRYP   | 000424 R   | #6-111      10-294  |
| \$ERRYQ   | 000464 R   | #6-112      10-285  |
| \$ERRYV   | 000524 R   | #6-113  |
| \$ERRYW   | 000562 R   | #6-114      9-242   |
| \$ERR1T   | = ***** GX | 8-208   |
| \$EXIT    | = 000000   | #12-319   |
| \$FAIL    | = 177777   | #12-319   |
| \$GPRM    | = *****    | 12-319  |
| \$LAMDA   | = 000000   | #12-319   |
| \$MUL     | = ***** GX | 7-164      7-171  |
| \$NUMBR   | = 000002   | #12-319   |
| \$RAD50   | = 000016   | #12-319   |
| \$RONLY   | = *****    | 12-319      12-319  |
| \$STRNG   | = 000004   | #12-319   |
| \$SUBXP   | = 000010   | #12-319   |
| \$\$\$FLG | = 177777   | #12-319   |
| \$\$\$KEY | = 177777   | #12-319   |
| .AMAXC    | = ***** GX | *10-291   |
| .AMAXH    | = ***** GX | *10-282   |
| .BFPTR    | = ***** GX | *7-130      7-137      *7-144      7-153  |
| .CCBNM    | = ***** GX | 5-86  |
| .CCBSZ    | = ***** GX | 5-87  |
| .MAXC     | = ***** GX | *9-222  |
| .MAXH     | = ***** GX | *9-229      9-230   |
| .MXPRO    | = ***** GX | 5-93      7-157      *7-159      7-161      7-168   |
| .NA       | = ***** GX | *9-254  |
| .NBEA     | = ***** GX | *10-274   |
| .NN       | = ***** GX | *9-218      9-245      10-265      11-298   |
| .NTPSZ    | = ***** GX | *7-187  |
| .PNUMB    | = ***** GX | *7-136      7-140      9-218      9-222      9-223      9-229      9-239      9-247      9-251        |
|           |            | 9-254      10-267      10-271      10-274      10-279      10-282      10-288      10-291      11-300 |
|           |            | 11-303      11-306  |
| .PNUMH    | = ***** GX | 9-237   |
| .RDBNM    | = ***** GX | 5-88      7-163   |
| .RDBSZ    | = ***** GX | 5-89      7-156   |
| .RDBTH    | = ***** GX | 5-92  |
| .ROUTB    | = ***** GX | *11-306   |
| .ROUTM    | = ***** GX | *9-239  |
| .SDBNM    | = ***** GX | 5-90      7-170   |
| .SDBSZ    | = ***** GX | 5-91      7-167   |
| .TPARS    | = ***** GX | 8-202   |

CF2AC4      CREATED BY MACRO ON 29-JUN-85 AT 03:11      PAGE 2      F 7

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|
| CALL    | 9-183 | 9-184 | 9-191 | 9-207 | 9-213 |
| EMSG\$R | #5-61 | 8-139 | 8-158 | 8-159 | 9-222 |
| FLTDF\$ | #5-61 | 5-63  |       |       |       |
| NTLRF\$ | #5-61 | 7-104 | 7-105 | 7-106 | 7-107 |
| RESMAP  | #5-72 | 9-214 |       |       |       |
| RETC    | #5-61 | 9-186 |       |       |       |
| RETURN  | 8-135 | 8-153 | 9-215 | 9-218 |       |
| SAVMAP  | #5-68 | 9-188 |       |       |       |
| SWSTK\$ | 9-183 |       |       |       |       |

```

320
321
322 .SBTTL .FLBK0 - FILE NAME BLOCK PROCESSING
323
324 .ENABL LSB
325
326
327
328
329
330
331
332
333
334
335
336
337 000502 012767 000040 000000G .FLBK0:MOV #S.SEC,SEROFF ; SAVE FILENAME LIST OFFSET
338 000510 052767 000400 000000G BIS #SF.SEC,SERFLG ; SET SECONDARY BOOT FILE FLAG
339 000516 000433 BR 5$ ; CONTINUE
340
341 000520 012767 000042 000000G .FLBK1:MOV #S.TER,SEROFF ; SAVE FILENAME LIST OFFSET
342 000526 052767 001000 000000G BIS #SF.TER,SERFLG ; SET TERTIARY BOOT FILE FLAG
343 000534 000424 BR 5$ ; CONTINUE
344
345 000536 012767 000044 000000G .FLBK2:MOV #S.LOA,SEROFF ; SAVE FILE NAME LIST OFFSET
346 000544 052767 002000 000000G BIS #SF.LOA,SERFLG ; SET LOAD FILE FLAG
347 000552 000415 BR 5$ ; CONTINUE
348
349 000554 012767 000046 000000G .FLBK3:MOV #S.DIA,SEROFF ; SAVE FILE NAME LIST OFFSET
350 000562 052767 004000 000000G BIS #SF.DIA,SERFLG ; SET DIAGNOSTIC FILE FLAG
351 000570 000406 BR 5$ ; CONTINUE
352
353 000572 012767 000050 000000G .FLBK4:MOV #S.DUM,SEROFF ; SAVE FILE NAME LIST OFFSET
354 000600 052767 010000 000000G BIS #SF.DUM,SERFLG ; SET UPLINE DUMP FILE FLAG
355
356 000606 5$: CALL SETBLK ; SET UP LOCAL COPY OF FNB
357 000612 CALL SRHFNB ; SEARCH FOR EXISTING FNB
358 000616 103006 BCC 10$ ; IF FOUND - BRANCH
359
360 000620 CALL CRFNBK ; IF NOT - CREATE NEW FNB
361 000624 103003 BCC 10$ ; IF ALLOCATION SUCCESSFUL - BRANCH
362
363 000626 EMSG$R FS ; FILE NAME BLOCK ALLOCATION FAILURE
364
365
366 000634 10$: SWSTK$ SETST ; ENTER SYSTEM STATE
367 000640 017746 000000G MOV @KSAR5,-(SP) ; SAVE MAPPING
368 000644 016646 000004 MOV 4(SP),-(SP) ; SET UNMAPPED ADDRESS OF FND (USER R0)
369 000650 CALL $CEACK ; GET ACCESS TO FNB
370 000654 012600 MOV (SP)+,R0 ; GET MAPPED ADDRESS
371 000656 105260 000002 INCB F,USE(R0) ; INCREMENT FNB USAGE COUNT
372
373 000662 016746 000000G MOV SERCUR,-(SP) ; SET UNMAPPED ADDRESS OF CURRENT SERVICE BLOCK
374 000666 CALL $CEACK ; ACCESS SERVICE BLOCK
375 000672 012600 MOV (SP)+,R0 ; GET MAPPED ADDRESS
376 000674 056760 000000G BIS SERFLG,S,FLAG(R0) ; SET UP FLAGS WORD

```

```

LOCAL DATA
73
79
80
81
82
83 000000
84
85
86
87
88 000000
89 000003
90 000004
91
92
93
94
95
96 000006
97 000010
98 000012
99 000014
100 000016

                .SBTTL  LOCAL DATA
                :***
                : LOCAL DATA
                :***

                .PSECT  DATA,D

                :
                : LINE IDENTIFICATION
                :
NAME:  .BLKB  3      ; DDM NAME
CTL:   .BLKB  1      ; CONTROLLER NUMBER
UNT:   .BLKB  1      ; UNIT NUMBER
       .EVEN

                :
                : STRING MANIPULATION FIELDS
                :
STRAD:  .BLKW  1      ; FIELD ADDRESS
STRMX:  .BLKW  1      ; MAXIMUM LENGTH
STRCT:  .BLKW  1      ; COUNT FIELD ADDRESS
STRPT:  .BLKW  1      ; POINTER TO '<' WITHIN RECORD BUFFER AREA
STRXX:  .BLKW  1      ; DUMMY COUNT FIELD

```

CF2  
MAC  
MAC  
ASL  
CAL  
DHB  
EMS  
NTL  
RES  
RES  
RET  
SAV  
SAV  
SLT  
SOB  
SWS

CF2AC6      CREATED BY MACRO ON 29-JUN-85 AT 00:12      PAGE 4      F 10

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES  |
|------------|---|
| ASL\$      | #5-57      8-244      8-259   |
| CALL       | 8-140      8-142      8-155      8-185      8-191      9-279      10-307                                  |
| DHBD\$     | #5-57      5-64   |
| EMSG\$R    | #5-57      8-235      8-271      9-291      10-337      10-384      11-396                                |
| NTLR\$     | #5-57      7-110      7-111      7-112      7-113      7-114      7-115                                   |
| RESMAP     | #5-57      #5-72  |
| RESRG      | #5-57      8-152  |
| RETURN     | 8-175      8-224      8-233      8-247      8-253      8-262      8-269      9-293      9-300      10-309 |
|            | 10-315      10-339      10-380      11-395  |
| SAVMAP     | #5-57      #5-68  |
| SAVRG      | #5-57      8-148  |
| SLTDF\$    | #5-57      5-63   |
| SOB        | 8-166      8-218      9-290      10-336   |
| SWSTK\$    | 8-140      8-185  |

|                |                |                  |                   |                  |
|----------------|----------------|------------------|-------------------|------------------|
| T\$LIFL 000013 | T\$LTIM 000026 | T\$SVC 000034    | X\$\$DBT= 000000  | .FUDGE= ***** GX |
| T\$LIFO 000013 | T\$LTPR 000014 | T\$T5 000030     | \$ERRXC= ***** GX | .HOME = ***** GX |
| T\$LIFS 000013 | T\$LTPS 000020 | T\$T6 000032     | \$ERRXE= ***** CX | .NA = ***** GX   |
| T\$LIN 000000  | T\$NAPL 000004 | T\$KMG= 000000   | \$ERRXJ= ***** GX | .NBEA = ***** GX |
| T\$LIPS 000006 | T\$NFE 000000  | T\$SMIN= 000000  | \$ERRXK= ***** GX | .NN = ***** GX   |
| T\$LLD 000012  | T\$NLEN 000010 | US\$CNF= 000002  | \$FUNC = ***** GX | .NOMAP= ***** GX |
| T\$LLDC 000045 | T\$NNUL 000002 | US\$DIS= 000006  | \$MAPX = ***** GX | .NTPSZ= ***** GX |
| T\$LLDL 000012 | T\$NOPL 000006 | US\$DON= 000000  | \$MUL = ***** GX  | .PCB = ***** GX  |
| T\$LLDO 000012 | T\$NRNI 000042 | US\$DSC= 000004  | \$UMRAL= ***** GX | .PNAM1= ***** GX |
| T\$LLDS 000012 | T\$NRPL 000005 | US\$EAC= 000012  | \$UMRPT= ***** GX | .PNAM2= ***** GX |
| T\$LLEN 000046 | T\$NRUL 000007 | US\$WDS= 000010  | \$SSHFT= 000001   | .PNUMB= ***** GX |
| T\$LOPR 000002 | T\$NVR 000001  | V\$SCTR= 001000  | .ALLOC= ***** GX  | .UMRFL= ***** GX |
| T\$LTCL 000024 | T\$RPR1 000040 | XPTLLN= ***** GX | .BRAD = ***** GX  |                  |

. ABS. 000124 000 (RW,1,GBL,ABS,OVR)  
 001206 001 (RW,1,LCL,REL,CDN)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 2  
 Work file writes: 4  
 Size of work file: 17167 Words ( 68 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:17.56  
 SY:CF2AC7.V2,[132,134]CF2AC7/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CF2AC7

```

252 .SBTTL PAS - SERVICE BLOCK PASSWORD
253
254
255 + PAS - SERVICE BLOCK PASSWORD
256
257 INPUTS -
258 .PSTPT - ADDRESS OF PARSED STRING
259 .PSTCN - SIZE OF STRING
260
261 OUTPUTS -
262 SERPAS - CONTAINS SERVICE BLOCK PASSWORD
263 -
264
265 000460 010446 PAS:: MOV R4,-(SP) ; SAVE REG
266 000462 010346 MOV R3,-(SP) ; SAVE TPARS REG
267 000464 012700 000000G MOV #HEXSTR,R0 ; GET BUFFER ADDRESS
268 000470 112720 000060 1$: MOVB #'0,(R0)+ ; FILL IN DEFAULT VALUES
269 000474 020027 000000G CMP R0,#ENDHEX ; END OF BUFFER ?
270 000500 001373 BNE 1$ ; IF NOT - BRANCH
271
272 000502 012700 000000G MOV #HEXSTR,R0 ; GET TEMPORARY BUFFER ADDRESS
273 000506 066700 000000G ADD .PSTCN,R0 ; STORE STRING IN REVERSE ORDER
274
275 000512 016701 000000G MOV .PSTCN,R1 ; GET SIZE OF STRING
276 000516 016702 000000G MOV .PSTPT,R2 ; GET STRING ADDRESS
277 000522 010104 MOV R1,R4 ; SAVE STRING LENGTH
278
279 000524 112240 10$: MOVB (R2)+,-(R0) ; STORE STRING IN WORK BUFFER
280 000526 005301 DEC R1 ; COUNT ONE LESS
281 000530 001375 BNE 10$ ; IF MORE - BRANCH
282 000532 012702 000000G MOV #SERPAS,R2 ; GET PASSWORD BUFFER ADDRESS
283 000536 012700 000000G MOV #HEXSTR,R0 ; SET START ADDRESS OF PASSWORD
284
285 000542 110312 20$: CALL HEXIND ; CONVERT FROM HEX
286 000546 MOV R3,(R2) ; STORE RESULT IN PASSWORD BUFFER
287 000550 CALL HEXIND ; CONVERT FROM HEX
288 000554 000241 CLC ; CLEAR C-BIT
289
290 000556 006303 ASL R3 ; POSITION VALUE
291 000560 006303 ASL R3 ; ...
292 000562 006303 ASL R3 ; ...
293 000564 006303 ASL R3 ; ...
294
295 000566 150322 BISB R3,(R2)+ ; STORE RESULT IN PASSWORD BUFFER
296 000570 022702 000000G CMP #ENDPAS,R2 ; CHECK FOR LAST CHARACTER
297 000574 001362 BNE 20$ ; IF MORE - BRANCH
298 000576 012603 MOV (SP)+,R3 ; RESTORE REG
299 000600 012604 MOV (SP)+,R4 ; RESTORE REG
300
301 000602 RETURN
302
303 +
304 ; HEXIND - CONVERT FROM HEX
305 -
306
307 000604 005003 HEXIND: CLR R3 ; INIT REG
308 000606 112003 MOVB (R0)+,R3 ; GET CHARACTER

```



.TITLE CF2AC9 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

3.00 14-APR-83  
PAR\$DF ACTION ROUTINES

4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0

5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

```
456
457 001066 STATES$ XPTDF
458 001066 TRANS$ %LLC$DF%
459
460 001066 STATES$
461 001066 TRANS$ %XPT%
462
463 001066 STATES$
464 001066 TRANS$ <','>
465
466 001066 STATES$ XPTDF1
467 001066 TRANS$ '! ,XPTDF1
468 001066 TRANS$ $RAD50,XPTDF1
469 001066 TRANS$ <','>
470
471 001066 STATES$
472 001066 TRANS$ $NUMBR
473
474 001066 STATES$
475 001066 TRANS$ <','>
476
477 001066 STATES$
478 001066 TRANS$ $NUMBR,$EXIT,XPCIR
479
480
481
```

CF2DEC - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 <sup>F 15</sup> 00:14 Page 5  
MACRO DEFINITIONS

```
45 .SBTTL MACRO DEFINITIONS
46
47
48 ; LOCAL MACROS
49 ;
50 .MCALL EMSG$,DHBDF$,ISTAT$,STATE$,TRANS$
51
52
53
54 ;+
55 ; OFFSET VALUES
56 ;-
57 000000 DHBDF$ ; DEFINE DECNET HOME BLOCK OFFSETS
58
59 000000 .PSECT
```

```

111 000132 110462 000000G      MOV  R4,M.BFVH(R2)      ;
112 000136                      SWSTK$ 40$      ;* ENTER KERNEL MODE
113 000142 010200              MOV  R2,R0      ;* COPY UMR BLOCK ADDRESS
114
115                      .IF DF R$$MPL
116
117                      MOV  @KSAR5,-(SP)      ;* SAVE CURRENT MAPPING
118                      MOV  ASUMR,-(SP)      ;* ASSUME SINGLE PROCESSOR SYSTEM
119                      TSTB @NCPU      ;* IS THIS A MULTI-PROCESSOR SYSTEM
120                      BEQ  25$      ;* IF EQ, NO
121                      MOV  6(SP),@KSAR5      ;* MAP TO CPU PARTITION (USER R0)
122                      MOV  #ASUM1,(SP)      ;* USE OUR OWN UMR ASSIGNMENT ROUTINE
123                      CALL @ (SP)+      ;* ASSIGN THE UMR'S
124                      MOV  (SP)+,@KSAR5      ;* RESTORE PREVIOUS MAPPING
125
126                      .IFF
127
128 000144                      CALL  $ASUMR      ;* TRY TO ASSIGN CONTIGUOUS UMR'S
129
130                      .ENDC
131
132 000150 103002 000010      BCC  30$      ;* IF CC, SUCCESS
133 000152 005066 000010      CLR  10(SP)      ;* INDICATE ERROR BY ZEROING USER MODE R3
134 000156                      RETURN      ;* BACK TO USER MODE
135 000160 005703      30$:    TST  R3      ;* WERE UMR'S ASSIGNED?
136 000162 001416      40$:    BEQ  111$      ;* IF ZERO, NO
137
138                      .IF DF R$$MPL
139                      TSTB @NCPU      ;* IS THIS A MULTIPROCESSOR SYSTEM ?
140
141                      BEQ  50$      ;* IF EQ, NO
142                      RETURN      ;* ELSE, LEAVE NOW
143
144                      50$:
145                      .ENDC
146
147 000164 010267 000000G      MOV  R2,,UMRBA      ;* SAVE UMR BLOCK ADDRESS
148 000170 016203 000000G      MOV  M.UMRA(R2),R3      ;* CALCULATE AND SAVE EVENTUAL $PUMR VALUE
149 000174 162703 170200      SUB  #UBMPR,R3      ;
150 000200 072327 000005      ASH  #5,R3      ;
151 000204 010367 000000G      MOV  R3,,PUMR      ;
152 000210 000415      BR  UMRLD      ;* LOAD UMR'S AND RETURN
153
154                      ;
155                      ; ERROR CONDITIONS
156
157 000212      101$:    MSG$R XL      ;* UMR BLOCK ALLOCATION FAILURE
158 000220      111$:    .IF DF R$$MPL
159
160                      TSTB @NCPU      ;* IS THIS A MULTI-PROCESSOR SYSTEM ?
161                      BNE  112$      ;* IF NE, YES .. LET NTLUNL DE-ALLOCATE
162
163                      .ENDC
164
165 000220 010200      MOV  R2,R0      ;* COPY UMR BLOCK ADDRESS
166 000222 012701 000000G      MOV  #M.LGTH,R1      ;* UMR BLOCK LENGTH
167 000226                      SWSTK$ 112$      ;* ENTER KERNEL MODE
168 000232                      CALLR @DEACB      ;* DE-ALLOCATE UMR BLOCK

```

**G 1.**

00:00

H 1

[illegible]

CFGX3

```

100                                     .SBTTL $QX3P - LOOK FOR X3P$DF MACRO
101
102                                     ;+
103                                     ; $QX3P - LOOK FOR X3P$DF MACRO
104                                     ;
105                                     ; INPUTS:
106                                     ;     NONE
107                                     ;
108                                     ; OUTPUTS:
109                                     ;     ALL REGISTERS DESTROYED
110                                     ;
111                                     ;-
112
113 000000 012705 000000' $QX3P:: MOV    #X3PDF,R5      ; STATE TABLE ADDRESS
114 000004 005001          CLR    R1                ; FULL KEYWORD MATCH LENGTH
115 000006 012702 000000'      MOV    #X3PKW,R2      ; KEYWORD TABLE ADDRESS
116 000012 016703 000000G      MOV    CFGSZ,R3       ; RECORD LENGTH
117 000016 012704 000000G      MOV    #CFGBF,R4      ; RECORD BUFFER ADDRESS
118 000022 005067 000000G      CLR    SYNERR        ; CLEAR SYNTAX ERROR FLAG
119 000026          CALL    ,TPARS                   ; GO DO THE PARSE
120 000032 103003          BCC    20$                ; IF CC, FOUND WHAT WE WERE LOOKING FOR
121 000034 005367 000000G      DEC    SYNERR        ; DID SYNTAX ERROR OCCUR?
122 000040 001401          BEQ    101$                ; IF EQ, YES
123 000042          20$:    RETURN
124
125                                     ; ERRORS
126
127 000044          101$:    MSG$R 1T                  ; SYNTAX ERROR

```

CF2A  
ERROR

CF2A

```

91                                     .SBTTL  ERROR MESSAGES
92
93                                     ;
94                                     ; ERROR MESSAGES
95                                     ;
96                                     .ENABL  LC
97                                     .NLIST  BEX
98
99 000012                             NTLERS$ ,YZ,8,CERR,RTSPC,,<Illegal area address>
100 000050                             NTLERS$ ,YY,8,CERR,RTSPC,,<Object or Remote block allocation failure>
101 000132                             NTLERS$ ,YX,8,CERR,RTSPC,,<Illegal node name>
102 000164                             NTLERS$ ,YV,8,CERR,RTSPC,,<Illegal node address>
103 000222                             NTLERS$ ,YN,8,CERR,RTSPC,CFLIN,<Illegal counter timer value>
104
105                                     .DSABL  LC
106                                     .LIST   BEX
107                                     .EVEN
108 000000                             .PSECT

```

CF2AC2 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 <sup>H 4</sup>00:10  
Table of contents

|    |    |                   |
|----|----|-------------------|
| 5- | 55 | MACRO DEFINITIONS |
| 6- | 75 | LOCAL DATA        |
| 7- | 91 | ERROR MESSAGES    |



.TITLE CF2AC3 - NTLCF2 ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 7-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

CF2AC3      CREATED BY    MACRO    ON 29-JUN-85 AT 00:10      PAGE 3      G 6

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |         |         |         |         |         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| ASL\$   | #5-59   |         |         |         |         |         |         |         |         |         |
| BFTAB\$ | #5-75   | 5-86    | 5-87    | 5-88    | 5-89    | 5-90    | 5-91    | 5-92    | 5-93    |         |
| CALL    | 7-152   | 7-164   | 7-171   | 8-202   |         |         |         |         |         |         |
| DBGTP\$ | #12-319 |         |         |         |         |         |         |         |         |         |
| EMSG\$  | #5-59   |         |         |         |         |         |         |         |         |         |
| EMSG\$R | #5-59   | 7-147   | 7-191   | 8-208   | 9-227   | 9-234   | 9-242   | 9-257   | 10-277  | 10-285  |
|         | 10-294  | 11-309  |         |         |         |         |         |         |         |         |
| ISTAT\$ | #5-59   | 12-319  |         |         |         |         |         |         |         |         |
| MTRAN\$ | #12-319 |         |         |         |         |         |         |         |         |         |
| NTLER\$ | #5-59   | 6-104   | 6-105   | 6-106   | 6-107   | 6-108   | 6-109   | 6-110   | 6-111   | 6-112   |
|         | 6-113   | 6-114   | 6-115   | 6-116   |         |         |         |         |         |         |
| RETURN  | 7-131   | 7-145   | 7-189   | 8-206   | 9-220   | 9-225   | 9-232   | 9-240   | 9-255   | 10-275  |
|         | 10-283  | 10-292  | 11-307  |         |         |         |         |         |         |         |
| SOB     | 7-181   |         |         |         |         |         |         |         |         |         |
| STATE\$ | #5-59   | 12-321  | #12-323 | #12-325 | #12-327 | #12-329 | #12-331 | #12-333 | #12-335 | #12-337 |
|         | #12-339 | #12-341 | #12-343 | #12-345 | #12-347 | #12-349 | #12-351 | #12-353 | #12-355 | #12-359 |
| TRANS   | #5-59   | #12-322 | #12-324 | #12-326 | #12-328 | #12-330 | #12-332 | #12-334 | #12-336 | #12-338 |
|         | #12-340 | #12-342 | #12-344 | #12-346 | #12-348 | #12-350 | #12-352 | #12-354 | #12-356 |         |

\*\*FILE\*\*ID\*\*CF2AC5

G 7

```
CCCCCCCC  FFFFFFFF  222222  AAAAAA  CCCCCCCC  5555555555
CCCCCCCC  FFFFFFFF  222222  AAAAAA  CCCCCCCC  5555555555
CC        FF        22        22  AA        AA  CC        55
CC        FF        22        22  AA        AA  CC        55
CC        FF        22        22  AA        AA  CC        555555
CC        FF        22        22  AA        AA  CC        555555
CC        FFFFFFFF  22        22  AA        AA  CC        55
CC        FFFFFFFF  22        22  AA        AA  CC        55
CC        FF        22        22  AAAAAAAA  CC        55
CC        FF        22        22  AAAAAAAA  CC        55
CC        FF        22        22  AA        AA  CC        55
CC        FF        22        22  AA        AA  CC        55
CCCCCCCC  FF        2222222222  AA        AA  CCCCCCCC  555555
CCCCCCCC  FF        2222222222  AA        AA  CCCCCCCC  555555
.....
.....
.....
.....
```

```
LL        SSSSSSSS  TTTTTTTTTT
LL        SSSSSSSS  TTTTTTTTTT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SSSSSS    TT
LL        SSSSSS    TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT
```

```

377 000702 066700 000000G ADD SEROFF,R0 ; POINT AT FILE NAME LIST
378 000706 016610 000004 MOV 4(SP),(R0) ; STORE UNMAPPED ADDRESS OF FILE BLOCK
379
380 000712 012677 000000G MOV (SP)+,@KSAR5 ; RESTORE MAPPING
381
382 000716 RETURN
383
384
385 000720 012767 000000G 000000G SETST:: MOV #FNBUFF,STRPT ; RESET START OF BUFFER
386 000726 005067 000000G CLR STRSIZ ; RESET STRING COUNT
387 000732 005067 000000G CLR FNBLN ; RESET FILE NAME LENGTH
388
389 000736 012700 000000G MOV #DEVBUF,R0 ; INIT DEVICE BUFFER
390 000742 012701 000000G MOV #DVSZ,R1 ; SIZE OF BUFFER
391 000746 CALL INIBUF ; INIT BUFFER
392
393 000752 012700 000000G MOV #UICBUF,R0 ; INIT UIC BUFFER
394 000756 012701 000000G MOV #UISZ,R1 ; SIZE OF BUFFER
395 000762 CALL INIBUF ; INIT BUFFER
396
397 000766 012700 000000G MOV #FILBUF,R0 ; INIT FILE NAME BUFFER
398 000772 012701 000000G MOV #FLSZ,R1 ; SIZE OF BUFFER
399 000776 CALL INIBUF ; INIT BUFFER
400
401 001002 RETURN
402
403 .DSABL LSB
404
405 000001 .END
  
```

```
102 .SBTTL ERROR MESSAGES
103
104 :: ERROR MESSAGES
105 ::
106 .ENABL LC
107 .NLIST BEX
108
109 NTLERS ,XN,8,CERR,RTSPC,CFLIN,<Process Not In System>
110 000020 NTLERS ,XP,8,CERR,RTSPC,CFLIN,<Line Not In System>
111 000056 NTLERS ,XT,8,CERR,RTSPC,CFLIN,<String Too Long>
112 000112 NTLERS ,XU,8,CERR,RTSPC,CFLIN,<Illegal node number>
113 000142 NTLERS ,YM,8,CERR,RTSPC,,<Station not in system>
114 000176 NTLERS ,L1,8,CERR,RTSPC,,<Illegal logging state>
115 000234
116
117 .LIST BEX
118 .DSABL LC
119 .EVEN
120 .PSECT
```

\*\*FILE\*\*ID\*\*CF2AC7

```

CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCC 77777777
CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCC 77777777
CC        FF        22        AA        AA    CC        77
CC        FF        22        AA        AA    CC        77
CC        FF        22        AA        AA    CC        77
CC        FF        22        AA        AA    CC        77
CC        FFFFFFFF 22        AA        AA    CC        77
CC        FFFFFFFF 22        AA        AA    CC        77
CC        FF        22        AAAAAA    CC        77
CC        FF        22        AAAAAA    CC        77
CC        FF        22        AA        AA    CC        77
CC        FF        22        AA        AA    CC        77
CCCCCCCC FF 2222222222 AA AA CCCCCC 77
CCCCCCCC FF 2222222222 AA AA CCCCCC 77

```

```

LL        SSSSSSSS TTTTTTTTTT
LL        SSSSSSSS TTTTTTTTTT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SSSSSS    TT
LL        SSSSSS    TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LL        SS        TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT

```

CF2AC7 CREATED BY MACRO ON 29-JUN-85 AT 00:12 PAGE 1 G 11  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES        |
|--------|------------|-------------------|
| ALOCB  | = ***** GX | 6-214             |
| ASLEN  | 000004     | 6-116             |
| CRSUB  | 000736 R   | 6-218 #7-344      |
| DEACB  | = ***** GX | 7-408             |
| ECLLN  | = ***** GX | 6-149             |
| ECLNOD | = ***** GX | 6-162             |
| ESNBR  | 000014     | #5-69             |
| ESNBS  | 000020     | #5-69             |
| ESNCR  | 000034     | #5-69             |
| ESNCS  | 000036     | #5-69             |
| ESNIC  | 000044     | #5-69             |
| ESNLEN | 000050     | #5-69 6-163       |
| ESNLLA | 000012     | #5-69             |
| ESNLNK | 000000     | #5-69             |
| ESNML  | 000040     | #5-69             |
| ESNMR  | 000024     | #5-69             |
| ESNMS  | 000030     | #5-69             |
| ESNNOD | 000002     | #5-69             |
| ESNRI  | 000042     | #5-69             |
| ESNRTP | 000005     | #5-69             |
| ESNSEG | 000010     | #5-69             |
| ESNTIM | 000046     | #5-69             |
| ESNUSE | 000004     | #5-69             |
| ESSTR  | 000006     | #5-69             |
| FE.EXT | = ***** GX | 6-177 6-258       |
| FE.PLA | = ***** GX | 6-268             |
| Fmask  | = ***** GX | 6-177 6-258 6-268 |
| IS\$AS | = *****    | 5-60              |
| K6TMP  | 000000 R   | #6-77             |
| LO.CON | 000006     | #5-68             |
| LO.CSR | 000006     | #5-68             |
| LO.CTL | 000014     | #5-68             |
| LO.LIN | 000004     | #5-68             |
| LO.PAR | 000010     | #5-68             |
| LO.PRI | 000010     | #5-68             |
| LO.SCR | 000014     | #5-68             |
| LO.STA | 000015     | #5-68             |
| LO.VCT | 000012     | #5-68             |
| LR.AST | 000026     | #5-68             |
| LR.CIR | 000002     | #5-68             |
| LR.CTL | 000004     | #5-68             |
| LR.EFN | 000004     | #5-68             |
| LR.LIN | 000002     | #5-68             |
| LR.PRM | 000030     | #5-68             |
| LR.PRO | 000002     | #5-68             |
| LR.STS | 000000     | #5-68             |
| LR.TCB | 000002     | #5-68             |
| LR.UNT | 000005     | #5-68             |
| LS.AST | = 100000   | #5-68             |
| LS.CEX | = 000001   | #5-68             |
| LS.CIR | = 000010   | #5-68             |
| LS.CXO | = 010000   | #5-68             |

CF2AC8 - CONFIG FILE ACTION ROLL MACRO V05.03b Saturday 29-Jun-85 00:12 Page 9-1

PAS - SERVICE BLOCK PASSWORD

|     |        |        |        |              |              |   |                             |
|-----|--------|--------|--------|--------------|--------------|---|-----------------------------|
| 309 | 000610 | 032703 | 000100 | BIT          | #LET,R3      | : | CHECK FOR LETTER            |
| 310 | 000614 | 001003 |        | BNE          | 10\$         | : | IF YES - BRANCH             |
| 311 | 000616 | 162703 | 000060 | SUB          | #'0,R3       | : | CONVERT TO BINARY VALUE     |
| 312 | 000622 | 000402 |        | BR           | 20\$         | : | CONTINUE                    |
| 313 | 000624 | 162703 | 000067 | 10\$: SUB    | #<'A-10.>,R3 | : | CONVERT HEX DIGIT TO BINARY |
| 314 | 000630 |        |        | 20\$: RETURN |              |   |                             |



CF2AC9 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:13 Page 5

MACRO DEFINITIONS

49  
50  
51  
52  
53  
54  
55

.SBTTL MACRO DEFINITIONS

;  
; LOCAL MACROS  
;

.MCALL EMSG\$,NTLERS\$,RETC,FNBDF\$,DHBDF\$,SAVMAP,RESMAP,SERDF\$  
.MCALL PDVDF\$,SLTDF\$,ISTAT\$,STAT\$,TRANS

```
483  
484  
485  
486  
487 001066  
488 001066  
489 001066  
490 001066  
491 001066  
492 001066  
493 001066  
494 001066  
495 001066  
496  
497 000001
```

```
      ;+  
      ;+ DEC$DF MACRO STATE TABLE FOR PHASE I INIT  
      ;+  
      STATES DECDEF  
      TRANS %DEC$DF%  
      STATES  
      TRANS $NUMBR,,ECLML  
      STATES  
      TRANS <','>  
      STATES  
      TRANS $NUMBR,$EXIT,ECLMC  
      STATES  
      .END
```

MACRO DEFINITIONS

```

61
62 000000 012705 000000' $0DEC:: MOV #DECDF,R5 ; SET UP STATE TABLE ADDRESS
63 000004 005001 CLR R1 ; FULL KEYWORD MATCH LENGTH
64 000006 012702 000000' MOV #DECKW,R2 ; KEYWORD TABLE ADDRESS
65 000012 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
66 000016 012704 000000G MOV #CFGBF,R4 ; BUFFER ADDRESS
67 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
68 000026 CALL TPARS ; PARSE LINE
69 000032 103003 BCC 20$ ; IF SUCCESS - BRANCH
70 000034 005367 000000G DEC SYNERR ; SYNTAX ERROR ?
71 000040 001401 BEQ 101$ ; IF YES - BRANCH
72 000042 20$: RETURN
73
74 000044 101$: MSG$R 1T ; SYNTAX ERROR
75

```

CF2UMR - NTL INITIALIZATION UMR MACRO V05.03b Saturday 29-Jun-85 00:14 Page 5-2

168 000236  
169

112\$: EMSG\$R XM  
.ENDC ; DF R\$11M

; NOT ENOUGH CONTIGUOUS UMR'S

CFGX2P CREATED BY MACRO ON 29-JUN-85 AT 00:08 PAGE 1 H 1  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES         |
|--------|------------|--------------------|
| CERR   | = ***** GX | 8-98 8-99 8-100    |
| CFGFB  | = ***** GX | 9-120              |
| CFGSZ  | = ***** GX | 9-119              |
| CFLIN  | = ***** GX | 8-98 8-99 8-100    |
| CHKDEV | = 000316 R | #11-225            |
| FMT8B  | = ***** GX | 8-98 8-99 8-100    |
| FM.8B  | = 000000   | #8-98 #8-99 #8-100 |
| ISSAS  | = *****    | 5-62               |
| LF.ACT | = 100000   | #5-63              |
| LF.BRO | = 000400   | #5-63              |
| LF.BWT | = 000007   | #5-63              |
| LF.ENA | = 002000   | #5-63              |
| LF.LPB | = 001000   | #5-63              |
| LF.MDC | = 000100   | #5-63              |
| LF.MFL | = 004000   | #5-63              |
| LF.MTP | = 000020   | #5-63              |
| LF.PAC | = 000200   | #5-63              |
| LF.RDY | = 040000   | #5-63              |
| LF.REA | = 010000   | #5-63              |
| LF.SER | = 000040   | #5-63              |
| LF.TIM | = 000010   | #5-63              |
| LF.UNL | = 020000   | #5-63              |
| LF.X2P | = 000000   | #5-63              |
| LN.CLO | = 000000   | #5-63              |
| LN.DUM | = 000005   | #5-63              |
| LN.LOA | = 000004   | #5-63              |
| LN.LOO | = 000003   | #5-63              |
| LN.OAU | = 000003   | #5-63              |
| LN.OFF | = 000001   | #5-63              |
| LN.ON  | = 000000   | #5-63              |
| LN.OOP | = 000004   | #5-63              |
| LN.OPE | = 000001   | #5-63              |
| LN.REF | = 000002   | #5-63              |
| LN.SER | = 000002   | #5-63              |
| LN.STA | = 000017   | #5-63              |
| LN.SUB | = 000360   | #5-63              |
| LN.TRI | = 000006   | #5-63              |
| L.COST | = 000015   | #5-63              |
| L.CTL  | = 000012   | #5-63 11-231       |
| L.CVA  | = 177776   | #5-63              |
| L.DDM  | = 000002   | #5-63 11-226       |
| L.DDS  | = 000004   | #5-63              |
| L.DLC  | = 000003   | #5-63              |
| L.DLM  | = 000006   | #5-63              |
| L.DLS  | = 000010   | #5-63              |
| L.FLG  | = 000000   | #5-63              |
| L.KRBA | = 000016   | #5-63              |
| L.LEN  | = 000022   | #5-63              |
| L.MPF  | = 000022   | #5-63              |
| L.NMST | = 000020   | #5-63              |
| L.NSTA | = 000014   | #5-63              |
| L.QWNR | = 000021   | #5-63              |

```

129          .SBTTL X3P$DF STATE TABLE
130          :
131          : TPARS STATE TABLES
132          :
133          : ISTAT$ X3PST,X3PKW
134          :
135          : X3P$DF
136          :
137          STATES$ X3PDF
138          TRANS$ %X3P$DF%,1,SYNERR
139
140          STATES$
141          TRANS$ $NUMBR,,X3.DBS ; DEFAULT BLOCK SIZE
142
143          STATES$
144          TRANS$ <','>
145
146          STATES$
147          TRANS$ $NUMBR,,X3.MBS ; MAXIMUM BLOCK SIZE
148
149          STATES$
150          TRANS$ <','>
151
152          STATES$
153          TRANS$ $NUMBR,,X3.DWS ; DEFAULT WINDOW SIZE
154
155          STATES$
156          TRANS$ <','>
157
158          STATES$
159          TRANS$ $NUMBR,,X3.MWS ; MAXIMUM WINDOW SIZE
160
161          STATES$
162          TRANS$ <','>
163
164          STATES$
165          TRANS$ $NUMBR,,X3.BEG ; BEGINNING OF TIMER VALUES
166
167          STATES$ X3NXT
168          TRANS$ $EOS,$EXIT,X3.END
169          TRANS$ <','>
170
171          STATES$
172          TRANS$ $NUMBR,X3NXT,X3.NXT ; NEXT TIMER VALUE
173
174          STATES$

```

CF2A  
ERROR

CF2A

```

110                                     .SBTTL  FIL$DF ACTION ROUTINES
111
112
113                                     ;+
114                                     GETSTR - STORE PARSED STRING
115
116                                     : INPUTS -
117                                     .PSTPT - ADDRESS OF PARSED STRING
118                                     .PSTCN - SIZE OF PARSED STRING
119                                     :-
120
121
122 000000 016700 000000G  GETSTR::MOV    .PSTPT,R0      ; GET ADDRESS OF STRING
123 000004 112077 000000G  10$:  MOV    (R0)+,R1      ; STORE BYTE
124 000010 005267 000000G      INC    STRPT      ; INC POINTER
125 000014 005267 000000G      INC    STRSIZ     ; COUNT ONE MORE CHARACTER
126 000020 005367 000000G      DEC    PSTCN     ; COUNT ONE LESS CHARACTER
127 000024 001367          BNE    10$          ; IF MORE - BRANCH
128
129 000026                                     RETURN
130
131
132                                     ;+
133                                     DEV - STORE DEVICE NAME
134
135                                     : INPUTS -
136                                     STRADD - CURRENT STRING ADDRESS
137                                     STRSIZ - STRING SIZE
138
139                                     : OUTPUTS -
140                                     DEVBUFF CONTAINS DEVICE SRING
141                                     :-
142
143 000030 016700 000000G  DEV::  MOV    STRADD,R0      ; GET STRING ADDRESS
144 000034 016701 000000G      MOV    STRSIZ,R1      ; GET STRING SIZE
145 000040 110167 000000G      MOV    R1,DEVBUF     ; STORE STRING SIZE
146 000044 012702 000001G      MOV    #DEVBUF+1,R2   ; GET STARTING ADDRESS OF STRING
147
148 000050 112022          1$:  MOV    (R0)+,(R2)+    ; STORE BYTE
149 000052 005301          DEC    R1                ; COUNT ONE LESS
150 000054 001375          BNE    1$                ; IF MORE - BRANCH
151 000056 112712 000072G      MOV    #'',(R2)      ; STORE COLON
152 000062 105267 000000G      INCB   DEVBUFF      ; COUNT THE :
153 000066 012767 000000G 000000G  MOV    #FN$BUFF,STRPT ; RESET START OF BUFFER ADDRESS
154 000074 005067 000000G      CLR    STRSIZ      ; INIT COUNT
155
156                                     RETURN
  
```

.TITLE CF2AC2 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0



```

53          .SBTTL  MACRO DEFINITIONS
54
55          ;
56          ; LIBRARY MACRO CALLS
57          ;
58          ;
59          .MCALL  EMSG$,EMSG$,NTLER$,ASL$,ISTAT$,TRANS$,STATES$
60
61
62
63          .SBTTL  LOCAL DATA
64
65          ;
66          ; LOCAL DATA
67          ;
68          ;
69
70          .PSECT  DATA,D
71
72          ;
73          ; BUFFER TABLE ENTRY
74          ;
75          .MACRO  BFTAB$  AA,BB
76          .IF  B  BB
77          .WORD  0
78          .IFF
79          .WORD  BB
80          .ENDC
81          .WORD  AA
82          .ENDM  BFTAB$
83
84          ; BUFFER PARAMETERS TABLE (ROUNDING FACTOR, ADDRESS)
85
86          BFTAB:  BFTAB$  .CCBNM
87                  BFTAB$  .CCBSZ,1
88                  BFTAB$  .RDBNM
89                  BFTAB$  .RDBSZ,1
90                  BFTAB$  .SDBNM
91                  BFTAB$  .SDBSZ,1
92                  BFTAB$  .RDBTH
93                  BFTAB$  .MXPRO
94
95          BFEND=.
```

\*\*FILE\*\*ID\*\*CF2AC4

H 6

```
CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCCCC 44 44
CCCCCCCC FFFFFFFF 222222 AAAAAA CCCCCCCC 44 44
CC FF 22 22 AA AA CC 44 44
CC FF 22 22 AA AA CC 44 44
CC FF 22 22 AA AA CC 44 44
CC FFFFFFFF 22 22 AA AA CC 4444444444
CC FFFFFFFF 22 22 AA AA CC 4444444444
CC FF 22 22 AAAAAAAAAA CC 44
CC FF 22 22 AAAAAAAAAA CC 44
CC FF 22 22 AA AA CC 44
CC FF 22 22 AA AA CC 44
CCCCCCCC FF 2222222222 AA AA CCCCCCCC 44
CCCCCCCC FF 2222222222 AA AA CCCCCCCC 44
.....
.....
.....
.....
```

```
LL SSSSSSSS TTTTTTTTTT
LL SSSSSSSS TTTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSS TT
LL SSSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

CF2AC5 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:11 <sup>H 7</sup>  
 Table of contents

|     |     |   |
|-----|-----|---|
| 5-  | 53  | MACRO DEFINITIONS                         |
| 8-  | 132 | CMPSTR - COMPARE FILE NAME BLOCK FIELDS   |
| 9-  | 191 | MOVBUF - SET UP LOCAL FILE NAME BLOCK     |
| 11- | 260 | CRFNBK - CREATE FILENAME DESCRIPTOR BLOCK |
| 12- | 321 | .FLBKO - FILE NAME BLOCK PROCESSING       |

CF2A  
 Symb

A\$\$C  
 A\$\$C  
 A\$\$P  
 A\$\$T  
 CERR  
 CFLI  
 CMPS  
 CRFN  
 C\$\$C  
 C\$\$O  
 C\$\$R  
 DECP  
 DEVB  
 DVSZ  
 D\$AM  
 D\$AM  
 D\$AN  
 D\$BR  
 D\$BR  
 D\$DE  
 D\$DE  
 D\$EN  
 D\$FN  
 D\$HI  
 D\$HC  
 D\$IN  
 D\$IN  
 D\$IF  
 D\$LI  
 D\$LM  
 D\$LM  
 D\$LS  
 D\$MA  
 D\$MA  
 D\$MA  
 D\$MI  
 D\$MN  
 D\$NA  
 D\$NE  
 D\$NE  
 D\$NE  
 D\$NI  
 D\$NP  
 D\$OU  
 D\$RI  
 D\$RI  
 D\$R  
 D\$SI  
 D\$SI  
 D\$SI  
 D\$SI

. AF  
 DAT  
 Err

```

Symbol table
A$$$CHK= 000000
A$$$CPS= 000000
A$$$PRI= 000000
A$$$TRP= 000000
CERR = ***** GX
CFLIN = ***** GX
CMPSTR 000116R
CRFNBK 000332R
C$$$CKP= 000000
C$$$ORE= 000400
C$$$RSH= 177564
DECPY = ***** GX
DEVBUF = ***** GX
DVSZ = ***** GX
D$AMXC 000072
D$AMXH 000074
D$ANN 000000
D$BRPR 000102
D$BRTM 000100
D$DEL 000045
D$DELEW 000046
D$END = 000104
D$FNB 000034
D$HIOR 000024
D$HOST 000022
D$INAC 000044
D$INCT 000042
D$IPL 000051
D$ID 000020
D$LNAM 000006
D$LNUM 000014
D$LST 000047
D$MAXC 000064
D$MAXH 000066
D$MAXV 000070
D$MLL 000040
D$MNOD 000041
D$NA 000062
D$NBEA 000056
D$NBRA 000054
D$NEND= 000054
D$NLN 000030
D$NN 000060
D$OUT 000043
D$RETF 000050
D$RNN 000002
D$RTMR 000076
D$SEG 000036
D$SER 000032
D$SRL 000052
D$BUG= 177514

D$$$ISK= 000000
D$$$L11= 000001
D$$$YNC= 000000
D$$$YNM= 000000
E$$$XPR= 000000
FIELD= ***** GX
FILBUF = ***** GX
FLSZ = ***** GX
FMT8 = ***** GX
FM.8 = 000000
FNBEXT= ***** GX
FNBLEN = ***** GX
FNBFSZ = ***** GX
FNBUFF = ***** GX
F$SLVL= 000001
F.NXT 000000
F.USE 000002
F.VAR 000003
G$$$TPP= 000000
G$$$TSS= 000000
G$$$TK= 000000
G$$$WRD= 000000
INIBUF 000160R
I$$$RAR= 000000
I$$$RDN= 000000
KSAR5 = ***** GX
K$$$CNT= 177546
K$$$CSR= 177546
K$$$LDC= 000000
K$$$TPS= 000074
LD$LP = 000000
LF.ACT= 100000
LF.BRO= 000400
LF.BWT= 000007
LF.ENA= 002000
LF.LPB= 001000
LF.MDC= 000100
LF.MFL= 004000
LF.MTF= 000020
LF.PAC= 000200
LF.RDY= 040000
LF.REA= 010000
LF.SER= 000040
LF.TIM= 000010
LF.UNL= 020000
LF.X2P= 000000
LN.CLO= 000000
LN.DUM= 000005
LN.LOA= 000004
LN.LOO= 000003
LN.OAU= 000003

LN.OFF= 000001
LN.ON= 000000
LN.OOP= 000004
LN.OPE= 000001
LN.REF= 000002
LN.SER= 000002
LN.STA= 000017
LN.SUB= 000360
LN.TRI= 000006
L$$$ASG= 000000
L$$$DRV= 000000
L$$$P11= 000001
L$$$11R= 000000
L.COST 000015
L.CTL 000012
L.CVA 177776
L.DDM 000002
L.DDS 000004
L.DLC 000003
L.DLM 000006
L.DLS 000010
L.FLG 000000
L.KRBA 000016
L.LEN = 000022
L.MPF 000022
L.NMST 000020
L.NSTA 000014
L.OWNR 000021
L.UNT 000013
MOVBUF 000174R
M$$$CRB= 000124
M$$$CRX= 000000
M$$$FCS= 000000
M$$$MGE= 000000
M$$$NET= 000000
M$$$OVR= 000000
N$$$ACC= 000001
N$$$BUF= 000001
N$$$LDV= 000001
N$$$MCP= 000001
N$$$MLL= 000001
N$$$MOV= 000010
N$$$NCT= 000001
N$$$PEM= 000001
P$$$P45= 000000
P$$$WRD= 000000
Q$$$OPT= 000010
REP8C = ***** GX
R$$$DER= 000000
R$$$K11= 000001

R$$$SND= 000000
R$$$11M= 000000
SERCUR= ***** GX
SERFLG= ***** GX
SEROFF= ***** GX
SETBLK 000216R
SETST 000720RG
SF.ACT= 000200
SF.CIR= 000010
SF.DEV= 000020
SF.DIA= 004000
SF.DPA= 000100
SF.DPC= 000200
SF.DUM= 010000
SF.ENA= 000100
SF.HAD= 000004
SF.HST= 000002
SF.LOA= 002000
SF.LPB= 000004
SF.MFL= 000040
SF.PAC= 000020
SF.PH3= 020000
SF.PSS= 000040
SF.REA= 000010
SF.SEC= 000400
SF.SER= 000001
SF.SVC= 000002
SF.TER= 001000
SF.UNL= 000040
SRHFN 000000R
STRPT = ***** GX
STRSZ= ***** GX
$$$WRG= 000000
$$$YSZ= 007600
S.ADD 000004
S.CIR 000016
S.COST 000001
S.DEV 000020
S.DIA 000046
S.DPA 000032
S.DPC 000036
S.DUM 000050
S.FLAG 000002
S.FLG 000000
S.HAD 000010
S.HST 000006
S.LEN 000004
S.LENG 000052
S.LNK 000000
S.LOA 000044

S.NMST 000002
S.OWNR 000003
S.PSS 000022
S.SEC 000040
S.TER 000042
T$$$KMG= 000000
T$$$MIN= 000000
UICBUF = ***** GX
UISZ = ***** GX
V$$$CTR= 001000
X$$$DBT= 000000
ZF.COY= 001000
ZF.DDM= 000001
ZF.DIA= 004000
ZF.DLC= 000002
ZF.DVP= 100000
ZF.INI= 040000
ZF.KMX= 000020
ZF.LLC= 000004
ZF.LMC= 000100
ZF.MAN= 020000
ZF.MFL= 000010
ZF.MTM= 000400
ZF.MUX= 000040
ZF.PSE= 002000
ZF.SLI= 010000
ZF.TIM= 000200
ZF.X3P= 000000
ZS.ASN= 100000
ZS.BSY= 140000
Z.AVL 000014
Z.DAT 000016
Z.DSP 000000
Z.FLG 000010
Z.LEN = 000016
Z.LLN 000006
Z.MAP 000020
Z.NAM 000004
Z.PCB 000012
Z.SCH 000007
Z$EACX= ***** GX
$ERRFS 000000R
$HEADR= ***** GX
$LINKX= ***** GX
$XALOC= ***** GX
.FLBK0 000502RG
.FLBK1 000520RG
.FLBK2 000536RG
.FLBK3 000554RG
.FLBK4 000572RG

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)
001004 001 (RW,I,LCL,REL,CON)
DATA 000046 002 (RW,I,LCL,REL,CON)
Errors detected: 0

```

```

122      ;
123      ; TPARS ACTION ROUTINES
124      ;
125      ;
126      ; START OF NODE NAME (NOD$DF)
127      ;
128      .ENABL  LSB
129
130      NODNM:: MOV  @DECPT,R0      ; GET ADDRESS OF DECNET HOME BLOCK
131              ADD  #D$LNAM,R0    ; ADDRESS OF NODE NAME
132              BEQ  10$           ; IF UNDEFINED, JUST EXIT
133              MOV  #6,R1         ; GET LENGTH
134              MOV  #STRXX,R2     ; NO COUNT FIELD
135              BR   10$
136
137      ;
138      ; START OF NODE IDENTIFICATION (NOD$DF)
139      ;
140      NODID:: SWSTK$ 30$         ; ENTER SYSTEM STATE
141              MOV  #34,,R1       ; GET NODE ID LENGTH
142      2$:      CALL  $XALOC      ; ALLOCATE NODE ID STRING
143              BCC  5$           ; IF SUCCESSFUL ALLOCATION - BRANCH
144              JOT          ; NO DSR LEFT
145
146      5$:      SUB  #4,R1         ; DO NOT INCLUDE COUNT FIELD IN STRING SIZE
147              MOV  @KSAR5,-(SP)  ; SAVE KISAR5 CONTENTS
148              <R1>          ; SAVE REG
149              SAVRG          ;
150              MOV  @DECPT,R1     ; GET HOME BLOCK ADDRESS
151              ADD  #D$LID,R1     ; POINT TO ID STRING ADDRESS
152              MOV  R0,(R1)       ; STORE UNMAPPED ADDRESS OF ID STRING
153              RESRG  <R1>       ; RESTORE SIZE
154
155      154:     MOV  R0,-(SP)      ; PUT UNMAPPED ADDRESS ON STACK
156      155:     CALL $CEACK       ; MAP TO POOL..
157      156:     MOV  (SP)+,R0     ; GET MAPPED ADDRESS
158
159      ;
160      ; THE FOLLOWING CODE WILL BE EXECUTED IN SYSTEM STATE FOR ID STRING PROCESSING
161      ;
162      162:     MOV  R0,STRAD      ; SAVE FIELD ADDRESS
163      163:     BEQ  25$          ; IF ZERO, JUST EXIT
164      164:     MOV  R1,STRMX     ; SAVE MAX LENGTH
165      165:     MOVB #1,(R0)+    ; BLANK FILL THE FIELD
166      166:     SOB  R1,20$      ;
167
168      168:     CMP  #32,,STRMX   ; NODE ID ?
169      169:     BNE  28$          ; IF NO - BRANCH
170      170:     MOV  (SP)+,@KSAR5 ; IF YES - RESTORE MAPPING
171
172      172:     MOV  .PSTPT,STRPT ; SAVE CURRENT TPARS POINTER
173      173:     INC  STRPT       ; BUT DON'T INCLUDE THE '<'
174
175      30$:     RETURN
176
177      .DSABL  LSB
178

```

CF2AC7 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:12 H 10  
Table of contents

CF2AC7 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:12  
Table of contents

```

5-    53    LOCAL MACROS
6-    75    DATA SECTION

```

CE2AC7 - CONF16 FILE ACTION ROLL MACRO V05.03b Saturday 29-Jun-85 00:12 Page 4

CE2AC7 - CONFIG FILE ACTION ROLL MACRO V05.03b Saturday 29-Jun-85 00:12 Page 4

CF2AC7 CREATED BY MACRO ON 29-JUN-85 AT 00:12 PAGE 2 H 11

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE       | REFERENCES |
|---------|-------------|------------|
| LS.CX5  | = 000400    | #5-68      |
| LS.DLM  | = 004000    | #5-68      |
| LS.ECH  | = 001000    | #5-68      |
| LS.FDX  | = 004000    | #5-68      |
| LS.HDX  | = 010000    | #5-68      |
| LS.LIN  | = 000003    | #5-68      |
| LS.LMC  | = 000007    | #5-68      |
| LS.NTI  | = 004000    | #5-68      |
| LS.ON   | = 000011    | #5-68      |
| LS.OPT  | = 000400    | #5-68      |
| LS.PRO  | = 000002    | #5-68      |
| LS.PWF  | = 040000    | #5-68      |
| LS.TOP  | = 002000    | #5-68      |
| LS.UNF  | = 020000    | #5-68      |
| LS.X25  | = 000012    | #5-68      |
| LS.11D  | = 004000    | #5-68      |
| LX.CEX  | = 000004    | #5-68      |
| LX.LIN  | = 000006    | #5-68      |
| LX.PRO  | = 000005    | #5-68      |
| L.LNG   | = 000124    | 6-150      |
| M\$MGE  | = 000000    | 6-85       |
| PARHD   | = ***** GX  | 6-193      |
| PLGTH   | = ***** GX  | 6-213      |
| PS.COM  | = ***** GX  | 7-363      |
| PS.NSF  | = ***** GX  | 7-363      |
| PS.SYS  | = ***** GX  | 6-205      |
| P.ALL   | = 000002 RG | #6-87      |
| P.ATT   | = ***** GX  | *6-271     |
| P.BLKS  | = ***** GX  | *7-352     |
| P.END   | = 000016 RG | #6-90      |
| P.MAIN  | = ***** GX  | *7-351     |
| P.NAM   | = ***** GX  | 6-198      |
| P.PRO   | = ***** GX  | *6-270     |
| P.REL   | = ***** GX  | 6-274      |
| P.STAT  | = ***** GX  | 6-205      |
| P.SUB   | = ***** GX  | 7-371      |
| R\$MPL  | = *****     | 6-229      |
| R\$11D  | = *****     | 5-60       |
| R\$11M  | = 000000    | 6-176      |
| T\$FLAG | = 000044    | #5-69      |
| T\$LIF  | = 000013    | #5-69      |
| T\$LIFL | = 000013    | #5-69      |
| T\$LIFO | = 000013    | #5-69      |
| T\$LIFS | = 000013    | #5-69      |
| T\$LIN  | = 000000    | #5-69      |
| T\$LIPS | = 000006    | #5-69      |
| T\$LLD  | = 000012    | #5-69      |
| T\$LLDC | = 000045    | #5-69      |
| T\$LLDL | = 000012    | #5-69      |
| T\$LLDO | = 000012    | #5-69      |
| T\$LLDS | = 000012    | #5-69      |
| T\$LLEN | = 000046    | #5-69      |

```

316
317
318
319
320
321
322
323
324
325
326
327
328 000632 012700 000000G SERCR::MOV #CIRNAM,R0 ; GET STRING STORAGE BUFFER
329 000636 016702 000000G MOV .PSTPT,R2 ; GET PARSED STRING ADDRESS
330 000642 016701 000000G MOV .PSTCN,R1 ; GET STRING COUNT
331 000646 112220 10$: MOV (R2)+,(R0)+ ; STORE DEVICE NAME
332 000650 005301 DEC R1 ; ANY LEFT ?
333 000652 001375 BNE 10$ ; IF YES - BRANCH
334
335 000654 RETURN
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350 000656 010146 FPDV: MOV R1,-(SP) ; STORE NAME TO BE MATCHED
351 000660 017701 000000G MOV @PDVTA,R1 ; GET PDV VECTOR ADDRESS
352 000664 017702 000000G MOV @PDVNM,R2 ; GET NUMBER OF PDV'S
353 000670 012100 10$: MOV (R1)+,R0 ; GET NEXT PDV ADDRESS
354 000674 026016 000004 CMP Z,NAM(R0),(SP) ; MATCH ?
355 000676 001405 BEQ 40$ ; IF YES - BRANCH
356 000700 005302 DEC R2 ; COUNT ONE LESS PDV
357 000702 001372 BNE 10$ ; IF MORE - BRANCH
358 000704 005726 TST (SP)+ ; IF NO MORE - ERROR
359 000706 000261 SEC ; SET C-BIT
360 000710 000404 BR 50$ ; REPORT ERROR
361
362 000712 005741 40$: TST -(R1) ; CALCULATE PDV INDEX
363 000714 167701 000000G SUB @PDVTA,R1 ; ...
364 000720 005726 TST (SP)+ ; CLEAN UP STACK
365
366 000722 50$: RETURN
367
368
369
370

```



```
57
58
59
60      ;+ OFFSET VALUES
61      ;-
62      000000      DHBDF$      ; DEFINE DECNET HOME BLOCK OFFSETS
63      000000      FNBDf$      ; DEFINE FILE NAME BLOCK OFFSETS
64      000000      SERDF$      ; DEFINE SERVICE BLOCK OFFSETS
65      000000      PDVDF$      ; DEFINE PDV OFFSETS
66      000000      SLTDF$      ; DEFINE SLT OFFSETS
67
68
69      ;+ LOCAL EQUIVS
70      ;-
71      000003      EXTN = 3      ;FILENAME EXTENSION MAXIMUM SIZE
72
73
74      000000      .PSECT
75
```

```

Symbol table
A$$$CHK= 000000 D$$$L11= 000001 LN.OPE= 000001 P.UMR 000410RG S.FLAG 000002
A$$$CPS= 000000 D$$$YNC= 000000 LN.REF= 000002 Q$$$OPT= 000010 S.FLG 000000
A$$$PRI= 000000 D$$$YNM= 000000 LN.SER= 000002 R$$$DER= 000000 S.FLG 000010
A$$$TRP= 000000 ECLLLN= ***** GX LN.STA= 000017 R$$$K11= 000001 S.HAD 000010
BRADF 000000R 002 ECLMC 001046RG LN.SUB= 000360 R$$$SND= 000000 S.HST 000006
BRADJ 000124R 002 ECLML 001036RG LN.TRI= 000006 R$$$11M= 000000 S.LEN 000004
BRADJ1 000134R 002 ECLNOD= ***** GX L$$$ASG= 000000 SCHEL 000116R 002 S.LENG 000052
BRDJ 001056RG EXT 000550RG L$$$DRV= 000000 SCHELO 000110R 002 S.LNK 000000
CFGRF = ***** GX EXTN = 000003 L$$$P11= 000001 SCTIM 000100R 002 S.LDA 000044
CFGSZ = ***** GX E$$$XPR= 000000 L$$$11R= 000000 SERDCT= ***** GX S.NMST 000002
CIPMUL= ***** GX FE.EXT= ***** GX L.COST 000015 S.OWNR 000003
CIRNUM= ***** GX FILBUF= ***** GX L.CTL 000012 S.PSS 000022
CIRNT= ***** GX FILE 000510RG L.CVA 177776 SERDMP= ***** GX S.SEC 000040
C$$$CKP= 000000 FMASK = ***** GX L.DDM 000002 SERDV 000762RG S.TER 000042
C$$$ORE= 000400 FNBUFF= ***** GX L.DDS 000004 SERHST= ***** GX T$$$KMG= 000000
C$$$RSH= 177564 F$$$LVL= 000001 L.DLC 000003 SERMUL 000662RG T$$$MIN= 000000
DADDR 001000RG F.NXT 000000 L.DLM 000006 SERMUX 000672RG V$$$CTR= 001000
DCT 001016RG F.USE 000002 L.DLS 000010 SERNOD= ***** GX XPCIR 001026RG
DECODEF 000174R 002 F.VAR 000003 L.DLG 000000 SERUNI 000652RG XPTDF 000142R 002
D$AMXC 000072 G$$$TPP= 000000 L.FLG 000000 SF.ACT= 000200 XPTDF1 000150R
D$AMXH 000074 G$$$TSS= 000000 L.KRBA 000016 SF.CIR= 000010 XPTLLN= ***** GX
D$ANN 000000 G$$$TTK= 000000 L.LEN = 000022 SF.DEV= 000020 X$$$DBT= 000000
D$BRPR 000102 G$$$TRD= 000000 L.MPF 000022 SF.DIA= 004000 ZF.COU= 001000
D$BRTM 000100 HSAD 000752RG L.NMST 000020 SF.DPA= 000100 ZF.DDM= 000001
D$DEL 000045 HSAR 000732RG L.NSTA 000014 SF.DPC= 000200 ZF.DIA= 004000
D$DELEW 000046 I$$$RAR= 000000 L.OWNR 000021 SF.DUM= 010000 ZF.DLC= 000002
D$END = 000104 I$$$RDN= 000000 L.UNT 000013 SF.ENA= 000100 ZF.DVP= 100000
D$FNB 000034 K$$$CNT= 177546 M$$$CRB= 000124 SF.HAD= 000004 ZF.INI= 040000
D$HIOB 000024 K$$$CSR= 177546 M$$$CRX= 000000 SF.HST= 000002 ZF.KMX= 000020
D$HOST 000022 K$$$LDC= 000000 M$$$CS= 000000 SF.LOA= 002000 ZF.LLC= 000004
D$INAC 000044 K$$$TPS= 000074 M$$$MGE= 000000 SF.LPB= 000004 ZF.LMC= 000100
D$INCT 000042 LD$LP = 000000 M$$$NET= 000000 SF.MFL= 000040 ZF.MAN= 020000
D$IPL 000051 LF.ACT= 100000 M$$$DVR= 000000 SF.PAC= 000020 ZF.MFL= 000010
D$LID 000020 LF.BRO= 000400 NOAD 000722RG SF.PH3= 020000 ZF.MTM= 000400
D$LNAM 000006 LF.BWT= 000007 NDAR 000702RG SF.PSS= 000040 ZF.MUX= 000040
D$LNUM 000014 LF.ENA= 002000 N$$$ACC= 000001 SF.REA= 000010 ZF.PSE= 002000
D$LST 000047 LF.LPB= 001000 N$$$BUF= 000001 SF.SEC= 000400 ZF.SLI= 010000
D$MAXC 000064 LF.MFC= 004000 N$$$LDV= 000001 SF.SER= 000001 ZF.TIM= 000200
D$MAXH 000066 LF.MDC= 000100 N$$$MCP= 000001 SF.SVC= 000002 ZF.X3P= 000000
D$MAXV 000070 LF.MFL= 000020 N$$$MLL= 000001 SF.TER= 001000 ZS.ASN= 100000
D$MLL 000040 LF.MTP= 000020 N$$$MOV= 000010 SF.UNL= 000040 ZS.BSY= 140000
D$MNDD 000041 LF.PAC= 000200 N$$$NCT= 000001 SLTDF1 000042R 002 Z.AVL 000014
D$NA 000062 LF.RDY= 040000 N$$$PEM= 000001 SLTDF2 000050R 002 Z.DAT 000016
D$NBEA 000056 LF.REA= 010000 PARHD = ***** GX SLTDF3 000024R 002 Z.DSP 000000
D$NBRA 000054 LF.SER= 000040 PNAME 000270RG STRADD= ***** GX Z.FLG 000010
D$NEND= 000054 LF.TIM= 000010 PS.CDM= ***** GX STRPT = ***** GX Z.LEN = 000016
D$NLN 000030 LF.UML= 020000 P$$$P45= 000000 STRSZ= ***** GX Z.LLN 000006
D$NN 000060 LF.X2P= 000000 P$$$WRD= 000000 SYNERR= ***** GX Z.MAP 000020
D$OUT 000043 LLLCKW 000000RG 003 P.BLKS= ***** GX Z.NAM 000004
D$RETF 000050 LN.CLO= 000000 002 P.BUSY= ***** GX Z.PCB 000012
D$RN 000002 LN.DUM= 000005 P.FUDG 000372RG S.ADD 000004 Z.SCH 000007
D$RTMR 000076 LN.LOA= 000004 P.MAIN= ***** GX $ALPHA= 000022
D$SEG 000036 LN.LOO= 000003 P.NAM = ***** GX $ANY = 000020
D$SER 000032 LN.OAU= 000003 P.NONE 000502RG S.DEV 000020 $BLANK= 000006
D$SQRL 000052 LN.OFF= 000001 P.PAR1 000066RG S.DIA 000046 $CAT5 = ***** GX
D$BUG= 177514 LN.ON = 000000 P.PAR2 000264RG S.DPA 000032 $DIGIT= 000024
D$ISK= 000000 LN.ODP= 000004 P.REL = ***** GX S.DPC 000036 $DNUMB= 000014
S.DUM 000050 P.STAT= ***** GX S.DUM 000050 $EOS = 000012

```

```

77
78
79
80
81
82
83
84 000052 105767 000001G ECLLNK::TSTB .PNUMB+1 ; > 255. ?
85 000056 001011 BNE 101$ ; IF YES - BRANCH
86 000060 017700 000000G MOV @DECPT,RO ; GET DECNET HOME BLOCK ADDRESS
87 000064 156760 000000G 000040 BISB .PNUMB,D$MLL(RO) ; STORE MAX LINKS IN HOME BLOCK
88 000072 016767 000000G 000000G MOV .PNUMB,$$LINX ; ...
89 000100 RETURN
90
91 000102 101$: MSG$R XO ; ILLEGAL MAX LOGICA LINKS VALUE
92
93 000110 105767 000001G NODCT:: TSTB .PNUMB+1 ; > 255. ?
94 000114 001011 BNE 101$ ; IF YES - BRANCH
95 000116 017700 000000G MOV @DECPT,RO ; GET DECNET HOME BLOCK ADDRESS
96 000122 156760 000000G 000041 BISB .PNUMB,D$MNOD(RO) ; STORE MAX NODE COUNTERS IN HOME BLOCK
97 000130 016767 000000G 000000G MOV .PNUMB,$$NODC ; ...
98 000136 RETURN
99
100 000140 101$: MSG$R XP ; ILLEGAL MAX NODE COUNTERS VALUE
101
102 000146 105767 000001G INCT:: TSTB .PNUMB+1 ; > 255. ?
103 000152 001006 BNE 101$ ; IF YES - BRANCH
104 000154 017700 000000G MOV @DECPT,RO ; GET DECNET HOME BLOCK ADDRESS
105 000160 156760 000000G 000042 BISB .PNUMB,D$INCT(RO) ; STORE INCOMING TIMER VALUE
106 000166 RETURN
107
108 000170 101$: MSG$R XU ; ILLEGAL INCOMING TIMER VALUE
109
110 000176 105767 000001G OUTT:: TSTB .PNUMB+1 ; > 255. ?
111 000202 001006 BNE 101$ ; IF YES - BRANCH
112 000204 017700 000000G MOV @DECPT,RO ; GET DECNET HOME BLOCK ADDRESS
113 000210 156760 000000G 000043 BISB .PNUMB,D$OUTT(RO) ; STORE OUTGOING TIMER VALUE
114 000216 RETURN
115
116 000220 101$: MSG$R XY ; ILLEGAL OUTGOING TIMER VALUE
117
118 000226 105767 000001G INAC:: TSTB .PNUMB+1 ; > 255. ?
119 000232 001006 BNE 101$ ; IF YES - BRANCH
120 000234 017700 000000G MOV @DECPT,RO ; GET HOME BLOCK ADDRESS
121 000240 156760 000000G 000044 BISB .PNUMB,D$INAC(RO) ; STORE INACTIVITY TIMER
122 000246 RETURN
123
124 000250 101$: MSG$R XW ; ILLEGAL INACTIVITY TIMER
125
126 000256 105767 000001G DELF:: TSTB .PNUMB+1 ; > 255. ?
127 000262 001006 BNE 101$ ; IF YES - BRANCH
128 000264 017700 000000G MOV @DECPT,RO ; GET HOME BLOCK ADDRESS
129 000270 156760 000000G 000045 BISB .PNUMB,D$DELF(RO) ; STORE DELAY FACTOR
130 000276 RETURN
131
132 000300 101$: MSG$R XQ ; ILLEGAL DELAY FACTOR VALUE
133

```

```

171
172
173
174
175
176
177
178
179
180
181
182
183 000244
184 000250 016203 000000G
185 000254 116204 000000G
186 000260 016205 000000G
187 000264 010500
188 000266 066200 000000G
189 000272 010325
190 000274 010425
191 000276 052703 020000
192 000302 025504
193 000304 020500
194 000306 103771
195 000310
196

;+
; UMRD - LOAD UNIBUS MAPPING REGISTERS
;
; INPUTS:
;   R2 = UMR BLOCK ADDRESS
;
; OUTPUTS:
;   UMR'S LOADED
;   R0,R3,R4,R5 = DESTROYED
;-
; IF DF R$$$11M & M$$$MGE
UMRLD: SWSTK$ 20$ ; ENTER KERNEL MODE
; GET PARTITION ADDRESS
MOV M.BFVL(R2),R3
MOVB M.BFVH(R2),R4
MOV M.UMRA(R2),R5 ; AND UMR ADDRESS ON I/O PAGE
MOV R5,R0 ; COPY UMR ADDRESS
ADD M.UMRN(R2),R0 ; GIVING UMR LIMIT
10$: MOV R3,(R5)+ ; LOAD A UMR
MOV R4,(R5)+ ; DOUBLEWORD REGISTER
ADD #20000,R3 ; INCREASE ADDRESS BY 4K WORDS
;
; ADC R4
; CMP R5,R0 ; DONE ?
; BLO 10$ ; IF LO, NO
20$: RETURN
.ENDC

```

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|---|
| L. UNT  | = 000013    | #5-63   |
| PDVTA   | = ***** GX  | 11-227  |
| RTSPC   | = ***** GX  | 8-98 8-99 8-100                                   |
| RS\$11D | = *****     | 5-62  |
| RS\$11M | = 000000    | 5-62  |
| RS\$11S | = *****     | 5-62  |
| SF. ACT | = 000200    | #5-63   |
| SF. ENA | = 000100    | #5-63   |
| SF. LPB | = 000004    | #5-63   |
| SF. MFL | = 000040    | #5-63   |
| SF. PAC | = 000020    | #5-63   |
| SF. REA | = 000010    | #5-63   |
| SF. SER | = 000001    | #5-63   |
| SF. SVC | = 000002    | #5-63   |
| SF. UNL | = 000040    | #5-63   |
| SYNERR  | = ***** GX  | *9-121 *9-124 *11-235 8-99 8-100 8-100            |
| S\$BAS  | = *****     | 8-98  |
| S. COST | = 000001    | #5-63   |
| S. FLG  | = 000000    | #5-63   |
| S. LEN  | = 000004    | #5-63   |
| S. NMST | = 000002    | #5-63   |
| S. OWN  | = 000003    | #5-63   |
| UCNT    | = ***** GX  | 11-233 11-250 11-261 11-273 11-287 11-301 *11-304 |
| X\$MBCB | = *****     | 5-62  |
| X2CTL   | = 000002 R  | #7-88 *11-214 11-231                              |
| X2DEV   | = 000000 R  | #7-87 *11-205 11-229                              |
| X2h MX  | = 001130    | #6-80 11-299                                      |
| X2F     | = 000000 R  | 9-116   |
| X2FKw   | = 000000 RG | 9-118 #10-136                                     |
| X2PST   | = 000000 RG | #10-136   |
| X2RTMX  | = 001130    | #6-79 11-285                                      |
| X2UNT   | = 000003 R  | #7-89 *11-215 11-233                              |
| X2WNMN  | = 000001    | #6-78 11-248                                      |
| X2WNMY  | = 000007    | #6-77 11-246                                      |
| X2. CTL | = 000244 R  | #11-210   |
| X2. DEV | = 000216 R  | #11-201   |
| X2. HLD | = 000614 R  | #11-295   |
| X2. MBS | = 000454 R  | #11-258   |
| X2. MRT | = 000510 R  | #11-269   |
| X2. MWS | = 000404 R  | #11-243   |
| X2. RTR | = 000500 R  | #11-281   |
| X2. UNT | = 000274 R  | #11-220   |
| ZF. COU | = 001000    | #5-62   |
| ZF. DDM | = 000001    | #5-62   |
| ZF. DIA | = 004000    | #5-62   |
| ZF. DLO | = 000002    | #5-62   |
| ZF. DVP | = 000000    | #5-62   |
| ZF. INI | = 040000    | #5-62   |
| ZF. KMX | = 000020    | #5-62   |
| ZF. LLC | = 000004    | #5-62   |
| ZF. LMC | = 000100    | #5-62   |
| ZF. MAN | = 020000    | #5-62   |

```

176                                     .SBTTL X3P$DF ACTION ROUTINES
177
178                                     ;
179                                     ; DEFAULT BLOCK SIZE
180
181 000052                                X3.DBS: CALL   CHKBLK           ; CHECK FOR LEGAL DEFAULT BLOCK SIZE
182 000056      103403                    BCS    101$           ; BR IF ERROR
183 000060      010067 000000G           MOV    RO,$X3DBS       ; STORE DEFAULT BLOCK SIZE
184 000064                                RETURN
185 000066      101$: MSG$R 2B           ; ILLEGAL DEFAULT BLOCK SIZE
186
187                                     ;
188                                     ; MAXIMUM BLOCK SIZE
189
190 000074                                X3.MBS: CALL   CHKBLK           ; CHECK FOR LEGAL MAXIMUM BLOCK SIZE
191 000100      103406                    BCS    101$           ; BR IF ILLEGAL VALUE
192 000102      020067 000000G           CMP    RO,$X3DBS       ; IS DEFAULT IN RANGE?
193 000106      103406                    BLO    102$           ; BR IF NO
194 000110      010067 000000G           MOV    RO,$X3MBS       ; SAVE MAXIMUM BLOCK SIZE
195 000114                                RETURN
196 000116      101$: MSG$R 2C           ; ILLEGAL MAXIMUM BLOCK SIZE
197 000124      102$: MSG$R 2D           ; DEFAULT BLOCK SIZE EXCEEDS MAXIMUM
198
199                                     ;
200                                     ; DEFAULT WINDOW SIZE
201
202 000132                                X3.DWS: CALL   CHKWND           ; CHECK FOR LEGAL WINDOW SIZE
203 000136      103403                    BCS    101$           ; BR IF ILLEGAL SIZE - ERROR
204 000140      010067 000000G           MOV    RO,$X3DWS       ; STORE DEFAULT WINDOW SIZE
205 000144                                RETURN
206 000146      101$: MSG$R 2E           ; ILLEGAL DEFAULT WINDOW SIZE
207
208                                     ;
209                                     ; MAXIMUM WINDOW SIZE
210
211 000154                                X3.MWS: CALL   CHKWND           ; CHECK FOR LEGAL WINDOW SIZE
212 000160      103406                    BCS    101$           ; BR IF ERROR
213 000162      020067 000000G           CMP    RO,$X3DWS       ; IS DEFAULT IN RANGE?
214 000166      103406                    BLO    102$           ; BR IF NO - ERROR
215 000170      010067 000000G           MOV    RO,$X3MWS       ; STORE MAXIMUM WINDOW SIZE
216 000174                                RETURN
217 000176      101$: MSG$R 2F           ; ILLEGAL MAXIMUM WINDOW SIZE
218 000204      102$: MSG$R 2G           ; DEFAULT WINDOW SIZE EXCEEDS MAXIMUM
219
220                                     ;
221                                     ; BEGINNING OF TIMER VALUES
222
223 000212      012767 00000G 000000' X3.BEG: MOV    #$X3CAL,NEXT ; POINT TO CALL TIMER VALUE
224 000220      105067 000010'          CLR    TIMRCT          ; CLEAR COUNTER
225
226                                     ;
227                                     ; NEXT TIMER VALUE
228
229 000224      105267 000010'          X3.NXT: INCB   TIMRCT          ; UPDATE COUNT
230 000230      126727 000010' 000007 CMPB   TIMRCT,#X3TMCT      ; MAXIMUM EXCEEDED?
231 000236      101015                    BHI    101$           ; BR IF YES - ERROR
232 000240      005767 000000G           TST    ,PNUMH          ; DOUBLE WORD VALUE?
233 000244      001017                    BNE    102$           ; BR IF YES - ERROR
234 000246      105767 000001G           TSTB   ,PNUMB+1        ; BYTE VALUE?
235 000252      001014                    BNE    102$           ; BR IF NO - ERROR
236 000254      016777 000000G 000000' MOV    ,PNUMB,@NEXT      ; STORE TIMER VALUE
237 000262      062767 000002 000000' ADD     #2,NEXT          ; POINT TO STORAGE FOR NEXT VALUE

```

```

158
159
160      ;+ GRPUIC - STORE GROUP CODE IN UICBUF
161      ;
162      ; INPUTS -
163      ; STRADD - PARSED STRING ADDRESS
164      ; STRSZ - STRING SIZE
165      ;
166      ; OUTPUTS -
167      ; GROUP STRING STORED IN UICBUF
168      ;
169      ; -
170
171 000102 016701 000000G GRPUIC::MOV STRSZ,R1 ; GET STRING SIZE
172 000106 016700 000000G MOV STRADD,R0 ; GET STRING ADDRESS
173
174 000112 110167 000000G MOVB R1,UICBUF ; STORE GROUP UIC STRING SIZE
175 000116 012702 000001G MOV #UICBUF+1,R2 ; GET POINTER TO BUFFER
176 000122 112722 000133 MOVB #'[(R2)+ ; STORE BRACKET
177 000126 105267 000000G INCB UICBUF ; COUNT BRACKET
178
179 000132 112022 1$: MOVB (R0)+,(R2)+ ; STORE STRING
180 000134 005301 DEC R1 ; COUNT ONE LESS
181 000136 001375 BNE 1$ ; IF MORE - BRANCH
182
183 000140 012767 000000G 000000G 10$: MOV #FNBUFF,STRPT ; RESET BUFFER POINTER
184 000146 005067 000000G CLR STRSZ ; INIT SIZE
185
186 000152 RETURN
187
188
189      ;+ USRUIC - STORE USER CODE IN UICBUF
190      ;
191      ; INPUTS -
192      ; STRADD - PARSED STRING ADDRESS
193      ; STRSZ - STRING SIZE
194      ;
195      ; OUTPUTS -
196      ; USER UIC CODE STORED IN UICBUF
197      ;
198      ; -
199 000154 052767 000001 000010' USRU1::BIS #1,NAMED ; SET NAMED DIRECTORY FLAG
200 000162 016701 000000G USRUIC::MOV STRSZ,R1 ; GET SIZE
201 000166 016700 000000G MOV STRADD,R0 ; GET ADDRESS
202 000172 005767 000010' TST NAMED ; NAMED DIRECTORY ?
203 000176 001405 BEQ 3$ ; IF NO - BRANCH
204 000200 112767 000133 000001G MOVB #'[UICBUF+1 ; LOAD [ CHAR
205 000206 105267 000000G INCB UICBUF ; COUNT BRACKET
206
207 000212 012702 000000G 3$: MOV #UICBUF,R2 ; GET BUFFER ADDRESS
208 000216 005046 CLRB -(SP) ;
209 000220 116716 000000G MOVB UICBUF,(SP) ; GET OLD UIC SIZE
210 000224 062602 ADD (SP)+,R2 ; ADJUST BUFFER POINTER
211
212 000226 005202 INC R2 ; INCLUDE SIZE BYTE
213 000230 032767 000001 000010' BIT #1,NAMED ; NAMED DIRECTORY ?
214 000236 001004 BNE 1$ ; IF YES - BRANCH
  
```

MACRO DEFINITIONS

```
55 .SBTTL MACRO DEFINITIONS
56
57 ;****
58 ; LIBRARY MACRO CALLS
59 ;****
60 .MCALL OBJDF$,EMSG$R,RETC,NTLER$
61
62 000000 OBJDF$ ; DEFINE OBJECT BLOCK OFFSETS
63
64 ;***
65 ; LOCAL MACRO DEFINITIONS
66 ;***
67 .MACRO SAVMAP
68 MOV @KSAR5,-(SP) ; SAVE APR 5
69 .ENDM
70
71 .MACRO RESMAP
72 MOV (SP)+,@KSAR5 ; RESTORE APR5
73 .ENDM
```



```

96                                     .SBTTL  ERROR MESSAGES
97
98                                     ;
99                                     ; ERROR MESSAGES
100                                    ;
101                                    .ENABL  LC
102                                    .NLIST  BEX
103
104 000040      NTLERS$ ,XR,8,CERR,RTSPC,CFLIN,<Too Many Arguments>
105 000074      NTLERS$ ,XS,8,CERR,RTSPC,CFLIN,<Too Few Arguments>
106 000126      NTLERS$ ,YK,8,CERR,RTSPC,,<Illegal cost value>
107 000162      NTLERS$ ,YL,8,CERR,RTSPC,,<Illegal hop value>
108 000214      NTLERS$ ,YM,8,CERR,RTSPC,,<Illegal maximum area value>
109 000260      NTLERS$ ,YN,8,CERR,RTSPC,,<Illegal broadcast router adjacency value>
110 000342      NTLERS$ ,YO,8,CERR,RTSPC,,<Illegal broadcast endnode adjacency value>
111 000424      NTLERS$ ,YP,8,CERR,RTSPC,,<Illegal area cost value>
112 000464      NTLERS$ ,YQ,8,CERR,RTSPC,,<Illegal area hop value>
113 000524      NTLERS$ ,YV,8,CERR,RTSPC,,<Illegal node address>
114 000562      NTLERS$ ,YW,8,CERR,RTSPC,,<Illegal routing timer interval>
115 000632      NTLERS$ ,YI,8,CERR,RTSPC,,<Illegal broadcast timer value>
116 000700      NTLERS$ ,YJ,8,CERR,RTSPC,,<Illegal router priority value>
117
118                                     .LIST  BEX
119                                     .DSABL LC
120                                     .EVEN
121 000000      .PSECT

```

CF2AC4 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:11<sup>1 6</sup>  
Table of contents

|    |     |  |
|----|-----|--|
| 5- | 54  | MACRO CALLS  |
| 6- | 77  | LOCAL DATA   |
| 7- | 96  | ERROR MESSAGES                                       |
| 8- | 114 | FLTD\$ ACTION ROUTINES FOR EVENT CLASS AND TYPE      |
| 9- | 161 | ACTION ROUTINES TO SET UP EVENT FILTER CONTROL BLOCK |

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51

.TITLE CF2AC5 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

.. COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
.. DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

.. THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
.. ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
.. INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
.. COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
.. OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
.. TRANSFERRED.

.. THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
.. AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
.. CORPORATION.

.. DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
.. SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

.. MODULE DESCRIPTION:

.. NTL - CONFIG FILE ACTION ROUTINES

.. DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

.. IDENT HISTORY:

- 1.00 12-JAN-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

CF2AC5 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:11 Page 12-3  
Symbol table

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15528 Words ( 61 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:16.40  
SY:CF2AC5.V2,[132,134]CF2AC5/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CF2AC5

```

179                                     ;
180                                     ; END-OF-STRING CHARACTER
181
182 000154 016702 000000G STRCH:: MOV ,PSTPT,R2 ; GET RIGHT BRACKET ADDRESS
183 000160 022767 000040 000010' CMP #32,,STRMX ; NODE ID STRING ?
184 000166 001017 BNE 5$ ; IF NO - BRANCH
185 000170 SWSTK$ 20$ ; IF YES - ENTER SYSTEM STATE
186 000174 017746 000000G MOV @KSAR5,-(SP) ; SAVE MAPPING
187 000200 017700 000000G MOV @DECP7,R0 ; GET HOME BLOCK ADDRESS
188 000204 062700 000020 ADD #D$LD,R0 ; GET ID STRING ADDRESS
189 000210 011000 MOV (R0),R0 ;
190 000212 010046 MOV R0,-(SP) ; MAP TO POOL..
191 000214 CALL $CEACK ;
192 000220 012600 MOV (SP)+,R0 ; GET MAPPED ADDRESS TO ID STRING
193 000222 010067 000006' MOV R0,STRAD ; SET UP FOR COMMON CODE
194
195                                     ;
196                                     ; THE FOLLOWING CODE WILL EXECUTE IN SYSTEM STATE FOR ID STRING PROCESSING
197                                     ;
198
199 000226 016700 000006' 5$: MOV STRAD,R0 ; GET STRING ADDRESS
200 000232 001425 BEQ 15$ ; IF ZERO, DO NOTHING
201 000234 016701 000014' MOV STRPT,R1 ; GET LEFT ANGLE BRACKET ADDRESS
202
203 000240 160102 SUB R1,R2 ; GIVING STRING LENGTH
204 000242 001006 BNE 8$ ; IF NON ZERO - BRANCH
205 000244 022767 000040 000010' CMP #32,,STRMX ; ID STRING ?
206 000252 001023 BNE 20$ ; IF NO - BRANCH
207 000254 010210 MOV R2,(R0) ; SETUP NODE ID STRING LENGTH OF ZERO
208 000256 000413 BR 15$ ; CONTINUE
209
210 000260 026702 000010' 8$: CMP STRMX,R2 ; VALID LENGTH ?
211 000264 103417 BLO 101$ ; IF LO, NO
212
213 000266 022767 000040 000010' CMP #32,,STRMX ; ID STRING ?
214 000274 001001 BNE 10$ ; IF NOT - BRANCH
215 000276 010220 MOV R2,(R0)+ ; SAVE STRING COUNT
216
217 000300 112120 10$: MOVB (R1)+,(R0)+ ; COPY THE STRING INTO THE FIELD
218 000302 SOB R2,10$ ; ...
219
220 000306 022767 000040 000010' 15$: CMP #32,,STRMX ; ID STRING ?
221 000314 001002 BNE 20$ ; IF NO - BRANCH
222 000316 012677 000000G MOV (SP)+,@KSAR5 ; IF YES - RESTORE MAPPING
223
224 000322 20$: RETURN
225
226                                     ;
227                                     ; ERROR CONDITIONS
228                                     ;
229 000324 022767 000040 000010' 101$: CMP #32,,STRMX ; ID STRING ?
230 000332 001006 BNE 102$ ; IF NO - BRANCH
231 000334 012677 000000G MOV (SP)+,@KSAR5 ; IF YES - RESTORE MAPPING
232 000340 012766 000350' 000016 MOV #102$,16(SP) ; UPDATE RETURN USER MODE ADDRESS
233 000346 RETURN
234
235 000350 102$: MSG$R XT ; ERROR - STRING TOO LONG

```

.TITLE CF2AC7 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 12-JAN-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0

## SYMBOL CROSS REFERENCE

CREF    04.00

## SYMBOL    VALUE      REFERENCES

|         |            |        |        |        |        |        |        |       |
|---------|------------|--------|--------|--------|--------|--------|--------|-------|
| T\$LOPR | 000002     | #5-69  |        |        |        |        |        |       |
| T\$LTCL | 000024     | #5-69  |        |        |        |        |        |       |
| T\$LTIM | 000026     | #5-69  |        |        |        |        |        |       |
| T\$LTPR | 000014     | #5-69  |        |        |        |        |        |       |
| T\$LTPS | 000020     | #5-69  |        |        |        |        |        |       |
| T\$NAPL | 000004     | #5-69  |        |        |        |        |        |       |
| T\$NFE  | 000000     | #5-69  |        |        |        |        |        |       |
| T\$NLEN | 000010     | #5-69  |        |        |        |        |        |       |
| T\$NNUL | 000002     | #5-69  |        |        |        |        |        |       |
| T\$NOPL | 000006     | #5-69  |        |        |        |        |        |       |
| T\$NRNI | 000042     | #5-69  |        |        |        |        |        |       |
| T\$NRPL | 000005     | #5-69  |        |        |        |        |        |       |
| T\$NRUL | 000007     | #5-69  |        |        |        |        |        |       |
| T\$NVR  | 000001     | #5-69  |        |        |        |        |        |       |
| T\$RPR1 | 000040     | #5-69  |        |        |        |        |        |       |
| T\$SVC  | 000034     | #5-69  |        |        |        |        |        |       |
| T\$T5   | 000030     | #5-69  |        |        |        |        |        |       |
| T\$T6   | 000032     | #5-69  |        |        |        |        |        |       |
| XPTLLN  | = ***** GX | 6-112  | 6-134  |        |        |        |        |       |
| \$ERRXC | = ***** GX | 6-186  | 6-196  |        |        |        |        |       |
| \$ERRXE | = ***** GX | 6-207  |        |        |        |        |        |       |
| \$ERRXJ | = ***** GX | 6-216  |        |        |        |        |        |       |
| \$ERRXK | = ***** GX | 6-227  |        |        |        |        |        |       |
| \$FUNC  | = ***** GX | 7-380  | 7-391  |        |        |        |        |       |
| \$MAPX  | = ***** GX | 6-277  |        |        |        |        |        |       |
| \$MUL   | = ***** GX | 6-117  | 6-136  | 6-151  | 6-164  |        |        |       |
| \$UMRAL | = ***** GX | 6-267  |        |        |        |        |        |       |
| \$UMRPT | = ***** GX | 6-260  |        |        |        |        |        |       |
| .ALLOC  | = ***** GX | *6-87  | 6-88   | *6-159 |        |        |        |       |
| .BRAD   | = ***** GX | 6-115  |        |        |        |        |        |       |
| .FUDGE  | = ***** GX | *6-88  | *6-107 | *6-128 | *6-147 | *6-160 | *6-174 | 7-353 |
| .HOME   | = ***** GX | *6-275 |        |        |        |        |        |       |
| .NA     | = ***** GX | 6-92   |        |        |        |        |        |       |
| .NBEA   | = ***** GX | 6-114  |        |        |        |        |        |       |
| .NN     | = ***** GX | 6-90   | 6-108  |        |        |        |        |       |
| .NOMAP  | = ***** GX | *6-106 | *6-127 | *6-146 | *6-173 |        |        |       |
| .NTPSZ  | = ***** GX | 7-352  |        |        |        |        |        |       |
| .PCB    | = ***** GX | 6-180  | *6-225 |        |        |        |        |       |
| .PNAM1  | = ***** GX | 6-183  | 7-349  | 7-350  |        |        |        |       |
| .PNAM2  | = ***** GX | 6-184  | 6-191  |        |        |        |        |       |
| .PNUMB  | = ***** GX | 6-87   |        |        |        |        |        |       |
| .UMRFL  | = ***** GX | *6-179 | *6-262 |        |        |        |        |       |

```

372                                     .SBTTL  FSLT - FIND SLT ADDRESS
373
374                                     ;+
375                                     FSLT - FIND SLT ADDRESS
376
377                                     INPUTS -
378                                     R0 - DDM PDV INDEX
379                                     R1 - CONTROLLER NUMBRE
380                                     R2 - UNIT NUMBER
381
382                                     OUTPUTS -
383                                     SERSLT - CONTAINS SLT ADDRESS
384                                     R3 - SLN
385                                     :-
386
387 000724 010446      FSLT:  MOV    R4,-(SP)      ; SAVE REG
388 000726 005046      CLR    -(SP)              ; CLEAR A WORD ON STACK
389 000730 110116      MOV    R1,(SP)            ; SET CONTROLLER NUMBER
390 000732 110266      MOV    R2,1(SP)           ; SET UNIT NUMBER
391 000736 017704      MOV    @SLTMA,R4          ; GET SLT VECTOR
392 000742 012703      MOV    #-1,R3            ; INIT SLN COUNTER
393
394 000746 012402      10$:  MOV    (R4)+,R2      ; GET NEXT SLT ADDRESS
395 000750 005203      INC    R3                  ; COUNT ONE MORE LINE
396 000752 127703      CMP    @SLTNM,R3          ; END OF SLT'S ?
397 000756 001002      BNE    15$                ; IF YES - ERROR
398 000760 000261      SEC                     ; SET C-BIT
399 000762 000414      BR     30$                ; CONTINUE
400
401 000764 126200      15$:  CMP    L,DDM(R2),R0  ; MATCH ?
402 000770 001366      BNE    10$                ; IF NO - BRANCH
403 000772 126216      CMP    L,CTL(R2),(SP)     ; CORRECT CONTROLLER ?
404 000776 001363      BNE    10$                ; IF NOT - BRANCH
405 001000 126266      CMP    L,UNT(R2),1(SP)    ; CORRECT UNIT ?
406 001006 001357      BNE    10$                ; IF NOT - BRANCH
407
408 001010 010267      MOV    R2,SERSLT          ; STORE SLT ADDRESS
409
410 001014 005726      30$:  TST    (SP)+         ; CLEAN STACK
411 001016 012604      MOV    (SP)+,R4          ; RESTORE REG
412
413 001020                                     RETURN
414
415
416                                     .END
  
```



```

77
78
79
80
81
82
83
84 000000 012705 000000' $QBRA:: .ENABL LSB ; SET UP STATE TABLE ADDRESS
85 000004 000405          BR #BRADF,R5 10$
86
87 000006 012705 000142' $QXPT:: MOV #XPTDF,R5 ; SET UP STATE TABLE ADDRESS
88 000012 000402          BR 10$ ; CONTINUE
89
90 000014 012705 000174' $QMLMC::MOV #DECDEF,R5 ; SET UP STATE TABLE ADDRESS
91
92 000020 005001 10$: CLR R1 ; FULL KEYWORD MATCH LENGTH
93 000022 012702 000000' MOV #LLCKW,R2 ; KEYWORD TABLE ADDRESS
94 000026 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
95 000032 012704 000000G MOV #CFGBF,R4 ; BUFFER ADDRESS
96 000036 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
97 000042          CALL TPARS ; PARSE LINE
98 000046 103003 BCC 20$ ; IF SUCCESS - BRANCH
99 000050 005367 000000G DEC SYNERR ; SYNTAX ERROR ?
100 000054 001401 BEQ 101$ ; IF YES - BRANCH
101 000056          20$: RETURN
102
103 000060          101$: MSG$R 1T ; SYNTAX ERROR
104
105          .DSABL LSB
106
107          .SBTTL ACTION ROUTINES FOR PAR$DF
108
109
110
111          ; TPARS ACTION ROUTINES
112
113
114          ; FIRST PARTITION NAME (PAR$DF)
115
116
117 000066 012702 000000G P.PAR1::MOV #.PNAM1,R2 ; POINT AT WORK AREA
118 000072          CALL PNAME ; RETRIEVE PARTITION NAME
119
120          .IF DF R$$11M
121
122 000076 016701 000000G MOV PARHD,R1 ; POINT AT SYSTEM LIST
123 000102 005000          CLR R0 ; INDICATE NO ERROR YET
124 000104          SWSTK$ 60$ ; ENTER KERNEL MODE
125 000110 011101 10$: MOV (R1),R1 ; GET NEXT PCB FROM LIST
126 000112 001443          BEQ 50$ ; IF ZERO, END OF LIST
127 000114 026112 000000G CMP P.NAM(R1),(R2) ; CHECK THE NAME
128 000120 001373          BNE 10$ ; TWO WORDS
129 000122 026162 000002G 000002 CMP P.NAM+2(R1),2(R2) ;
130 000130 001367          BNE 10$ ;
131
132          .IF DF R$$MPL
133

```

I 14  
CF2AC9 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:13 Page 12-2  
Symbol table

|                   |                  |                    |                  |                  |
|-------------------|------------------|--------------------|------------------|------------------|
| \$ERRXA= ***** GX | \$LAMDA= 000000  | \$SUBXP= 000010    | .NAMBF= ***** GX | .PNUMB= ***** GX |
| \$ERRXB= ***** GX | \$NUMBR= 000002  | \$\$\$FLG= 177777  | .NOMAP= ***** GX | .PNUMH= ***** GX |
| \$ERRXD= ***** GX | \$QBRA 000000RG  | \$\$\$KEY= 000006  | .NTPSZ= ***** GX | .PSTCN= ***** GX |
| \$ERRXF= ***** GX | \$QMLMC 000014RG | \$\$\$STA= 000000  | .PCB = ***** GX  | .PSTPT= ***** GX |
| \$ERRIT= ***** GX | \$QXPT 000006RG  | \$\$\$TMP= 000050R | .PNAM1= ***** GX | .TPARS= ***** GX |
| \$EXIT = 000000   | \$RAD50= 000016  | .BRAD = ***** GX   | .PNAM2= ***** GX | .UMRFL= ***** GX |
| \$FAIL = 177777   | \$STRNG= 000004  | .FUDGE= ***** GX   |                  |                  |

|         |        |     |                    |
|---------|--------|-----|--------------------|
| . ABS   | 177776 | 000 | (RW,I,GBL,ABS,OVR) |
|         | 001066 | 001 | (RW,I,LCL,REL,CON) |
| \$STATE | 000212 | 002 | (RW,D,LCL,REL,CON) |
| \$KTAB  | 000016 | 003 | (RW,D,LCL,REL,CON) |
| \$KSTR  | 000057 | 004 | (RW,D,LCL,REL,CON) |

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 2  
Work file writes: 3  
Size of work file: 16954 Words ( 67 Pages)  
Size of core pool: 17608 Words ( 67 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:01:01.04  
SY:CF2AC9.V2,[132,134]CF2AC9/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CF2AC9

```

134 000306 105767 000001G DELW:: TSTB .PNUMB+1 ; > 255. ?
135 000312 001006 BNE 101$ ; IF YES - BRANCH
136 000314 017700 000000G MOV @DECPT,R0 ; GET HOME BLOCK ADDRESS
137 000320 156760 000000G 000046 BISB .PNUMB,D$DELW(R0) ; STORE DELAY WEIGHT
138 000326 RETURN
139
140 000330 101$: MSG$R XR ; ILLEGAL DELAY WEIGHT VALUE
141
142 000336 105767 000001G IPL:: TSTB .PNUMB+1 ; > 255. ?
143 000342 001006 BNE 101$ ; IF YES - BRANCH
144 000344 017700 000000G MOV @DECPT,R0 ; GET HOME BLOCK ADDRESS
145 000350 156760 000000G 000051 BISB .PNUMB,D$IPL(R0) ; STORE INPUT PACKET LIMITER
146 000356 RETURN
147
148 000360 101$: MSG$R XS ; ILLEGAL INPUT PACKET LIMITER VALUE
149
150
151 000366 105767 000001G LST:: TSTB .PNUMB+1 ; > 255. ?
152 000372 001006 BNE 101$ ; IF YES - BRANCH
153 000374 017700 000000G MOV @DECPT,R0 ; GET HOME BLOCK ADDRESS
154 000400 156760 000000G 000047 BISB .PNUMB,D$LST(R0) ; STORE LINK SERVICE THRESHOLD
155 000406 RETURN
156
157 000410 101$: MSG$R XT ; ILLEGAL LINK SERVICE THRESHOLD VALUE
158
159
160 000416 105767 000001G RETF:: TSTB .PNUMB+1 ; > 255. ?
161 000422 001006 BNE 101$ ; IF YES - BRANCH
162 000424 017700 000000G MOV @DECPT,R0 ; GET HOME BLOCK ADDRESS
163 000430 156760 000000G 000050 BISB .PNUMB,D$RETF(R0) ; STORE RETRANSMIT FACTOR
164 000436 RETURN
165
166 000440 101$: MSG$R XX ; ILLEGAL RETRANSMIT FACTOR VALUE
167
168 000446 017700 000000G SEG:: MOV @DECPT,R0 ; GET HOME BLOCK ADDRESS
169 000452 016760 000000G 000036 MOV .PNUMB,D$SEG(R0) ; STORE ECL SEGMENT SIZE
170 000460 RETURN
171

```

198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254

```

*** - $ASUM1 - ASSIGN UNIBUS MAPPING REGISTERS

THIS ROUTINE IS CALLED TO ASSIGN A CONTIGUOUS SET OF UMR'S. NOTE THAT
FOR THE SAKE OF SPEED, THE LINK WORD OF EACH MAPPING ASSIGNMENT BLOCK
POINTS TO THE UMR ADDRESS (2ND) WORD OF THE BLOCK, NOT THE FIRST WORD.
THE CURRENT STATE OF UMR ASSIGNMENTS IS REPRESENTED BY A LINKED LIST OF
MAPPING ASSIGNMENT BLOCKS, EACH BLOCK CONTAINING THE ADDRESS OF THE
FIRST UMR ASSIGNED AND THE NUMBER OF UMR'S ASSIGNED TIMES 4. THE BLOCKS
ARE LINKED IN THE ORDER OF INCREASING FIRST UMR ADDRESS.

THIS ROUTINE SHOULD BE CALLED FROM SYSTEM STATE WITH KISAR5 MAPPED TO
THE LOWCR (LOW CORE) FOR THIS CPU.

NOTE - THIS ROUTINE ASSUMES THAT LOWCR IS MAPPED BY APR0.

INPUTS:
R0 = POINTER TO A MAPPING REGISTER ASSIGNMENT BLOCK.
M.UMRN(R0) = THE NUMBER OF UMR'S REQUIRED * 4.

OUTPUTS:
R4,R5 = DESTROYED
C=0 IF THE UMR'S WERE SUCCESSFULLY ASSIGNED.
ALL FIELDS OF THE MAPPING REGISTER ASSIGNMENT BLOCK
ARE INITIALIZED AND THE BLOCK IS LINKED INTO THE
ASSIGNMENT LIST.

C=1 IF THE UMR'S COULD NOT BE ASSIGNED.

-

      .IF DF R$$MPL
$ASUM1:
      MOV    UMRPT,R4      ; GET $UMRPT ADDRESS
      ADD    #120000,R4    ; BIAS BY APR5
      MOV    R0,(R4)       ; STORE UMR BLOCK ADDRESS
      TST    (R0)+         ; SKIP OVER LINK WORD
      MOV    UMRHD,R5      ; GET $UMRHD VALUE
      ADD    #120002,R5    ; BIAS BY APR5
10$:   MOV    R5,R4         ; SAVE POINTER TO PREVIOUS
      BEQ    30$           ; IF EQ, ASSIGNMENT FAILURE
      MOV    (R5),(R0)     ; POINT TO NEXT FREE UMR
      ADD    M.UMRN-M.UMRA(R5),(R0) ;
      MOV    #UBMPR<31.*4>,-(SP) ; CALCULATE END OF UMR'S
      MOV    -(R5),R5      ; POINT TO NEXT ASSIGNMENT BLOCK
      BEQ    20$           ; IF EQ, THERE IS NONE
      MOV    (R5),(SP)     ; SET NEXT ALLOCATED UMR ADDRESS
20$:   SUB    (R0),(SP)     ; CALCULATE NUMBER OF FREE UMR'S * 4
      CMP    (SP)+,M.UMRN-M.UMRA(R0) ; ARE THERE ENOUGH UMR'S AVAILABLE ?
      BLO    10$           ; IF LO, NO
      MOV    R0,-(R4)      ; LINK NEW BLOCK INTO CHAIN
      MOV    (R0),R4       ; SAVE FIRST UMR ADDRESS
      MOV    R5,-2(R0)     ; POINT NEW BLOCK TO NEXT IN LIST
      MOV    R4,R5        ; DUPLICATE FIRST UMR ADDRESS
      BICB   #UBMPR:34,R5  ; MASK OUT ALL BUT HI 2 BITS IN LOW BYTE
      BIC    R5,R4         ; CLEAR OUT HIGH BYTE AND HIGH 2 BITS
      ASR    R5            ; SHIFT HIGH 2 BITS TO BITS 4 & 5

```

| SYMBOL   | VALUE    | REFERENCES |        |        |        |        |        |        |               |
|----------|----------|------------|--------|--------|--------|--------|--------|--------|---------------|
| ZF.MFL   | = 000010 | #5-62      |        |        |        |        |        |        |               |
| ZF.MTM   | = 000400 | #5-62      |        |        |        |        |        |        |               |
| ZF.MUX   | = 000040 | #5-62      |        |        |        |        |        |        |               |
| ZF.PSE   | = 002000 | #5-62      |        |        |        |        |        |        |               |
| ZF.SLI   | = 010000 | #5-62      |        |        |        |        |        |        |               |
| ZF.TIM   | = 000200 | #5-62      |        |        |        |        |        |        |               |
| ZF.X3P   | = 000000 | #5-62      |        |        |        |        |        |        |               |
| ZS.ASN   | = 100000 | #5-62      |        |        |        |        |        |        |               |
| ZS.BSY   | = 140000 | #5-62      |        |        |        |        |        |        |               |
| Z.AVL    | 000014   | #5-62      |        |        |        |        |        |        |               |
| Z.DAT    | 000016   | #5-62      |        |        |        |        |        |        |               |
| Z.DSP    | 000000   | #5-62      | 5-62   |        |        |        |        |        |               |
| Z.FLG    | 000010   | #5-62      |        |        |        |        |        |        |               |
| Z.LEN    | = 000016 | #5-62      |        |        |        |        |        |        |               |
| Z.LLN    | 000006   | #5-62      |        |        |        |        |        |        |               |
| Z.MAP    | 000020   | #5-62      |        |        |        |        |        |        |               |
| Z.NAM    | 000004   | #5-62      | 11-229 |        |        |        |        |        |               |
| Z.PCB    | 000012   | #5-62      |        |        |        |        |        |        |               |
| Z.SCH    | 000007   | #5-62      |        |        |        |        |        |        |               |
| \$ALPHA  | = 000022 | #10-136    |        |        |        |        |        |        |               |
| \$ANY    | = 000020 | #10-136    |        |        |        |        |        |        |               |
| \$BLANK  | = 000006 | #10-136    |        |        |        |        |        |        |               |
| \$CAT5   | = *****  | GX 11-204  |        |        |        |        |        |        |               |
| \$DIGIT  | = 000024 | #10-136    |        |        |        |        |        |        |               |
| \$DNUMB  | = 000014 | #10-136    |        |        |        |        |        |        |               |
| \$EOS    | = 000012 | #10-136    |        |        |        |        |        |        |               |
| \$ERR1T  | = *****  | GX 9-130   |        |        |        |        |        |        |               |
| \$ERR2C  | = *****  | GX 11-265  |        |        |        |        |        |        |               |
| \$ERR2F  | = *****  | GX 11-254  |        |        |        |        |        |        |               |
| \$ERR2I  | = *****  | GX 11-238  |        |        |        |        |        |        |               |
| \$ERR2J  | 000004   | R #8-98    | 11-277 |        |        |        |        |        |               |
| \$ERR2K  | 000046   | R #8-99    | 11-291 |        |        |        |        |        |               |
| \$ERR2L  | 000106   | R #8-100   | 11-306 |        |        |        |        |        |               |
| \$EXIT   | = 000000 | #10-136    |        |        |        |        |        |        |               |
| \$FAIL   | = 177777 | #10-136    |        |        |        |        |        |        |               |
| \$GPRM   | = *****  | 10-136     |        |        |        |        |        |        |               |
| \$LAMDA  | = 000000 | #10-136    |        |        |        |        |        |        |               |
| \$NUMBER | = 000002 | #10-136    |        |        |        |        |        |        |               |
| \$QX2P   | 000144   | RG #9-116  |        |        |        |        |        |        |               |
| \$RAD50  | = 000016 | #10-136    |        |        |        |        |        |        |               |
| \$RONLY  | = *****  | 10-136     | 10-136 |        |        |        |        |        |               |
| \$SLTA   | = *****  | GX 11-225  |        |        |        |        |        |        |               |
| \$STRNG  | = 000004 | #10-136    |        |        |        |        |        |        |               |
| \$SUBXP  | = 000010 | #10-136    |        |        |        |        |        |        |               |
| \$X2HLD  | = *****  | GX *11-303 |        |        |        |        |        |        |               |
| \$X2MBS  | = *****  | GX *11-263 |        |        |        |        |        |        |               |
| \$X2MRT  | = *****  | GX *11-275 |        |        |        |        |        |        |               |
| \$X2MWS  | = *****  | GX *11-252 |        |        |        |        |        |        |               |
| \$X2RET  | = *****  | GX *11-289 |        |        |        |        |        |        |               |
| \$FLG    | = 177777 | #10-136    |        |        |        |        |        |        |               |
| \$KEY    | = 177777 | #10-136    |        |        |        |        |        |        |               |
| .PNUMB   | = *****  | GX 11-212  | 11-214 | 11-222 | 11-224 | 11-245 | 11-260 | 11-271 | 11-275 11-283 |

```

233 000270          005267 000000G      101$: RETURN          ; INDICATE SYNTAX ERROR
234 000272          062716 000002      INC      SYNERR          ; REJECT THE TRANSITION
235 000276          062716 000002      ADD      #2,(SP)
236 000302          000000          102$: RETURN          ; ILLEGAL TIMER VALUE
237 000304          000000          EMSG$R  2H
238          ;
239          ; END OF TIMER VALUES
240          ;
241 000312 126727 000010' 000007 X3.END: CMPB    TIMRCT,#X3TMCT    ; WERE ALL VALUES PROCESSED?
242 000320 001402          BEQ      10$          ; BR IF YES
243 000322 062716 000002          ADD      #2,(SP)          ; REJECT THE TRANSITION
244 000326          000001          10$: RETURN
245          .END

```

```

215
216 000240 112722 000054      MOVB  #'',(R2)+      ; STORE GROUP/USER UIC SEPARATOR
217 000244 105267 000000G     INCB  UICBUF        ; INCLUDE ,
218
219 000250 005067 000010'      1$: CLR  NAMED        ; INIT NAMED DIRECTORY FLAG
220 000254 112022              MOVB  (R0)+,(R2)+      ; STORE STRING
221 000256 105267 000000G     INCB  UICBUF        ; COUNT ONE CHARACTER
222 000262 005301              DEC   R1              ;
223 000264 001371              BNE   1$              ; IF MORE - BRANCH
224
225 000266 112712 000135      MOVB  #'',(R2)          ; STORE UIC TERMINATOR CHARACTER
226 000272 105267 000000G     INCB  UICBUF        ; ADJUST UIC STRING SIZE
227
228 000276 012767 000000G 000000G 10$: MOV  #FNBUFF,STRPT ; RESET PARSED STRING POINTER
229 000304 005067 000000G     CLR   STRSZ          ; INIT SIZE
230
231 000310                      RETURN
232
233

```

```

75          .SBTTL  LOCAL DATA
76          ;****
77          ; LOCAL DATA
78          ;****
79
80          .PSECT  DATA,D
81
82          ;
83          ; OBJECT BLOCK VALUES
84          ;
85          TYPE:  .BLKB  1          ; OBJECT TYPE NUMBER
86          FLAG:  .BLKB  1          ; OBJECT FLAGS
87          ONAME:  .BLKW  2          ; OBJECT NAME IN RAD50
88          MXCP:  .BLKB  1          ; MAXIMUM NUMBER OF COPIES TO SPAWN
89          .EVEN
  
```



```

123                                     .SBITL TPARS ACTION ROUTINES FOR BUF$DF
124
125                                     ;
126                                     ; TPARS ACTION ROUTINES
127                                     ;
128                                     ; START OF BUFFER DEFINITIONS (BUF$DF)
129                                     ;
130 000000 012767 000000' 000000G B.STR:: MOV #BFTAB,.BFPTR ; SETUP TABLE POINTER
131 000006                                     RETURN
132
133                                     ;
134                                     ; BUFFER DEFINITION NUMBER (BUF$DF)
135                                     ;
136 000010 005067 000000G B.ZERO:: CLR .PNUMB ; ZERO THE COUNTER
137 000014 016700 000000G B.NUM:: MOV .BFPTR,R0 ; GET BUFFER TABLE POINTER
138 000020 022700 000040' CMP #BFEND,R0 ; AT END OF TABLE ?
139 000024 001410 BEQ 101$ ; IF EQ, YES - TOO MANY ARGUMENTS
140 000026 016701 000000G MOV .PNUMB,R1 ; GET NUMBER
141 000032 061001 ADD (R0),R1 ; ROUND IT UP
142 000034 042001 BIC (R0)+,R1 ;
143 000036 010130 MOV R1,(R0)+ ; STORE THE NUMBER
144 000040 010067 000000G MOV R0,.BFPTR ; SAVE NEW POINTER VALUE
145 000044                                     RETURN
146
147 000046 101$: EMSG$R XR ; ERROR - TOO MANY ARGS
148
149                                     ;
150                                     ; END OF BUFFER DEFINITIONS (BUF$DF)
151                                     ;
152 000054 B.ZEND:: CALL B.ZERO ; ZERO THE LAST PARAMETER
153 000060 026727 000000G 000040' B.END:: CMP .BFPTR,#BFEND ; AT END OF TABLE ?
154 000066 001053 BNE 101$ ; IF NE, NO - TOO FEW ARGUMENTS
155
156 000070 016700 000000G 5$: MOV .RDBSZ,R0 ; GET RDB SIZE
157 000074 032767 000001 000000G BIT #1,.MXPRO ; ODD VALUE ?
158 000102 001402 BEQ 7$ ; IF NO - BRANCH
159 000104 005267 000000G INC .MXPRO ; MAKE VALUE EVEN
160
161 000110 066700 000000G 7$: ADD .MXPRO,R0 ; ADD DLC OVERHEAD
162 000114 062700 000000G ADD #CRC$OV,R0 ; ADD CRC OVERHEAD TO RDB SIZE
163 000120 016701 000000G MOV .RDBNM,R1 ; GET NUMBER OF RDB'S
164 000124 CALL $MUL ; GIVING BYTES NEEDED FOR RDB'S
165 000130 010146 MOV R1,-(SP) ; SAVE COUNT
166 000132 010046 MOV R0,-(SP) ; DOUBLE PRECISION
167 000134 016700 000000G MOV .SDBSZ,R0 ; GET SDB SIZE
168 000140 066700 000000G ADD .MXPRO,R0 ; ADD DLC OVERHEAD
169 000144 062700 000000G ADD #CRC$OV,R0 ; ADD CRC OVERHEAD
170 000150 016701 000000G MOV .SDBNM,R1 ; GET NUMBER OF SDB'S
171 000154 CALL $MUL ; GIVING BYTES NEEDED FOR SDB'S
172 000160 062600 ADD (SP)+,R0 ; INCLUDE RDB COUNT
173 000162 062601 ADD (SP)+,R1 ; DOUBLE PRECISION
174 000164 005500 ADC R0 ;
175 000166 062701 000077 ADD #77,R1 ; ROUND UP TO NEXT MULTIPLE OF 100
176 000172 005500 ADC R0 ;
177
178 000174 012702 000006 .IF DF M$SMGE ; SHIFT DOUBLE REGISTER
179 000200 006200 10$: ASR R0 ; RIGHT SIX PLACES

```

.TITLE CF2AC4 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

MACRO DEFINITIONS

53 .SBTTL MACRO DEFINITIONS  
54  
55  
56 : LOCAL MACROS  
57 :  
58 .MCALL EMSG\$,NTLR\$,RETC,FNBDF\$,DHBDF\$,SERDF\$  
59 .MCALL PDVDF\$,SLTDF\$

CF2A  
SYME  
SYME  
CERR  
CFLI  
CMP\$  
CRFN  
DECF  
DEVE  
DVSZ  
DSFN  
FIEL  
FILE  
FLSZ  
FMT\$  
FM.\$  
FNBE  
FNBL  
FNBS  
FNBU  
F.NX  
F.US  
F.VA  
INIE  
I\$S  
KSAF  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LF.  
LN.  
LN.  
LN.  
LN.  
LN.  
LN.  
LN.  
LN.  
LN.  
LN.  
LN.  
LN.

| SYMBOL | VALUE      | REFERENCES |        |        |         |                    |
|--------|------------|------------|--------|--------|---------|--------------------|
| CERR   | = ***** GX | 6-80       |        |        |         |                    |
| CFLIN  | = ***** GX | 6-80       |        |        |         |                    |
| CMPSTR | = 000116 R | 7-122      | #8-145 | 8-152  | 8-161   |                    |
| CRFNBK | = 000332 R | #11-273    | 12-360 |        |         |                    |
| DECP   | = ***** GX | 7-102      | 11-297 |        |         |                    |
| DEVBUF | = ***** GX | 10-238     | 10-240 | 12-389 |         |                    |
| DVSZ   | = ***** GX | 12-390     |        |        |         |                    |
| D\$FNB | = 000034   | 7-103      | 11-298 |        |         |                    |
| FIELD  | = ***** GX | 7-121      |        |        |         |                    |
| FILBUF | = ***** GX | 10-249     | 10-250 | 12-397 |         |                    |
| FLSZ   | = ***** GX | 12-398     |        |        |         |                    |
| FMT8   | = ***** GX | 6-80       |        |        |         |                    |
| FM.8   | = 000000   | #6-80      |        |        |         |                    |
| FNBEXT | = ***** GX | 10-253     |        |        |         |                    |
| FNBLEN | = ***** GX | *10-254    | 11-274 | 11-288 | *12-387 |                    |
| FNBZ   | = ***** GX | 10-232     |        |        |         |                    |
| FNBUFF | = ***** GX | 7-117      | 10-231 | 11-290 | 12-385  |                    |
| F.NXT  | = 000000   | #6-67      |        |        |         |                    |
| F.USE  | = 000002   | #6-67      | 7-118  | 7-119  | *12-371 |                    |
| F.VAR  | = 000003   | #6-67      |        |        |         |                    |
| INIBUF | = 000160 R | #8-182     | 10-233 | 12-391 | 12-395  | 12-399             |
| ISSAS  | = *****    | 6-69       |        |        |         |                    |
| KSAR5  | = ***** GX | 7-104      | 7-127  | 11-281 | 11-314  | 12-367      12-380 |
| LF.ACT | = 100000   | #6-70      |        |        |         |                    |
| LF.BRO | = 000400   | #6-70      |        |        |         |                    |
| LF.BWT | = 000007   | #6-70      |        |        |         |                    |
| LF.ENA | = 002000   | #6-70      |        |        |         |                    |
| LF.LPB | = 001000   | #6-70      |        |        |         |                    |
| LF.MDC | = 000100   | #6-70      |        |        |         |                    |
| LF.MFL | = 004000   | #6-70      |        |        |         |                    |
| LF.MTP | = 000020   | #6-70      |        |        |         |                    |
| LF.PAC | = 000200   | #6-70      |        |        |         |                    |
| LF.RDY | = 040000   | #6-70      |        |        |         |                    |
| LF.REA | = 010000   | #6-70      |        |        |         |                    |
| LF.SER | = 000040   | #6-70      |        |        |         |                    |
| LF.TIM | = 000010   | #6-70      |        |        |         |                    |
| LF.UNL | = 020000   | #6-70      |        |        |         |                    |
| LF.X2P | = 000000   | #6-70      |        |        |         |                    |
| LN.CLO | = 000000   | #6-70      |        |        |         |                    |
| LN.DUM | = 000005   | #6-70      |        |        |         |                    |
| LN.LCA | = 000004   | #6-70      |        |        |         |                    |
| LN.LOO | = 000003   | #6-70      |        |        |         |                    |
| LN.OAU | = 000003   | #6-70      |        |        |         |                    |
| LN.OFF | = 000001   | #6-70      |        |        |         |                    |
| LN.ON  | = 000000   | #6-70      |        |        |         |                    |
| LN.OOP | = 000004   | #6-70      |        |        |         |                    |
| LN.OPE | = 000001   | #6-70      |        |        |         |                    |
| LN.REF | = 000002   | #6-70      |        |        |         |                    |
| LN.SER | = 000002   | #6-70      |        |        |         |                    |
| LN.STA | = 000017   | #6-70      |        |        |         |                    |
| LN.SUB | = 000360   | #6-70      |        |        |         |                    |
| LN.TRI | = 000006   | #6-70      |        |        |         |                    |

```

236
237
238 ; STORE NODE/HOST NUMBER (NOD$DF)
239 ;
240 ;
241 000356 022767 000077 000000G NODARE::CMP .ENABL LSB ; CHECK FOR LEGAL VALUE
242 000364 002472 BLT 101$ ; IF ILLEGAL AREA - BRANCH
243 000366 016701 000000G MOV .PNUMB,R1 ; STORE VALUE IN A REGISTER
244 000372 ASL$ 10,R1 ; POSITION AREA BITS
245 000416 017700 000000G MOV @DECPT,R0 ; STORE DECNET HOME BLOCK POINTER
246 000422 010160 000014 MOV R1,D$LNUM(R0) ; STORE AREA NUMBER
247 000426 RETURN
248
249 000430 022767 001777 000000G NODNUM::CMP #1023...PNUMB ; CHECK FOR LEGAL NODE ADDRESS
250 000436 002445 BLT 101$ ; IF ILLEGAL - BRANCH
251 000440 017700 000000G MOV @DECPT,R0 ; STORE DECNET HOME BLOCK POINTER
252 000444 056760 000000G 000014 BIS .PNUMB,D$LNUM(R0) ; STORE NODE ADDRESS
253 000452 RETURN
254
255
256 000454 022767 000077 000000G HSTARE::CMP #63...PNUMB ; CHECK FOR LEGAL HOST AREA
257 000462 002433 BLT 101$ ; IF ILLEGAL - BRANCH
258 000464 016701 000000G MOV .PNUMB,R1 ; STORE VALUE IN REGISTER
259 000470 ASL$ 10,R1 ; POSITION AREA BITS
260 000514 017700 000000G MOV @DECPT,R0 ; STORE DECNET HOME BLOCK ADDRESS
261 000520 010160 000022 MOV R1,D$HOST(R0) ; STORE HOST AREA
262 000524 RETURN
263
264
265 000526 022767 001777 000000G HSTNUM::CMP #1023...PNUMB ; CHECK FOR LEGAL HOST NODE ADDRESS
266 000534 002406 BLT 101$ ; IF ILLEGAL - BRANCH
267 000536 017700 000000G MOV @DECPT,R0 ; STORE DECNET HOME BLOCK ADDRESS
268 000542 056760 000000G 000022 BIS .PNUMB,D$HOST(R0) ; STORE HOST ADDRESS
269 000550 RETURN
270
271 000552 ; 101$: MSG$R XU ; ERROR - ILLEGAL NODE NUMBER
272 .DSABL LSB

```

```
53          .SBTTL  LOCAL MACROS
54
55          ; LOCAL MACROS
56          ;
57          .MCALL  EMSG$R,NTLDF$,CTRDF$,LLTDF$,ADJDF$
58
59          .IF DF  R$$$11D ! 1$$$AS
60          .MCALL  MAP$$,CRRG$$,RDBDF$
61
62          RDBDF$          ; DEFINE RDB OFFSETS AND SYMBOLS
63
64          .ENDC
65
66          ADJDF$          ; DEFINE ADJACENCY DATA BASE SYMBOLS
67 000000          ; DEFINE NTL OFFSETS AND SYMBOLS
68 000000          ; DEFINE NTL OFFSETS AND SYMBOLS
69 000000          ; DEFINE COUNTER BLOCK OFFSETS
70 000000          ; LOGICAL LINK OFFSETS
71
72
```

CF2AC7      CREATED BY    MACRO    ON 29-JUN-85 AT 00:12      PAGE 4      J 11

MACRO CROSS REFERENCE

CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |       |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| ADJDF\$ | #5-58 | 5-67  |       |       |       |       |       |       |       |       |
| CALL    | 6-117 | 6-136 | 6-151 | 6-164 | 6-192 | 6-214 | 6-218 | 6-224 | 6-267 | 6-276 |
|         | 6-277 | 7-408 |       |       |       |       |       |       |       |       |
| CTRDF\$ | #5-58 | 5-69  |       |       |       |       |       |       |       |       |
| EMSG\$R | #5-58 | 6-186 | 6-227 |       |       |       |       |       |       |       |
| LLTDF\$ | #5-58 | 5-70  |       |       |       |       |       |       |       |       |
| NLTDF\$ | #5-58 | 5-68  |       |       |       |       |       |       |       |       |
| RETURN  | 6-221 | 6-281 | 7-402 | 7-410 |       |       |       |       |       |       |
| SOB     | 7-548 |       |       |       |       |       |       |       |       |       |
| SWSTK\$ | 6-192 | 6-276 |       |       |       |       |       |       |       |       |

|                                      |                  |                  |                  |                   |
|--------------------------------------|------------------|------------------|------------------|-------------------|
| AS\$CHK= 000000                      | D\$SYNC= 000000  | LN.REF= 000002   | SERBEG= ***** GX | S.HAD 000010      |
| AS\$CPS= 000000                      | D\$SYNM= 000000  | LN.SER= 000002   | SERCR 000632RG   | S.HST 000006      |
| AS\$PRI= 000000                      | ENDHEX= ***** GX | LN.STA= 000017   | SERCUR= ***** GX | S.LEN 000004      |
| AS\$TRP= 000000                      | ENDPAS= ***** GX | LN.SUB= 000360   | SERDCT= ***** GX | S.LENG 000052     |
| CERR = ***** GX                      | E\$XPR= 000000   | LN.TRI= 000006   | SERDEV= ***** GX | S.LNK 000000      |
| CFLIN = ***** GX                     | FMT8 = ***** GX  | L\$ASG= 000000   | SERDMP= ***** GX | S.LOA 000044      |
| CIRMUL= ***** GX                     | FM.8 = 000000    | L\$DRV= 000000   | SEREND 000144RG  | S.NMST 000002     |
| CIRNAM= ***** GX                     | FPDV 000656R     | L\$P11= 000001   | SERFIN= ***** GX | S.OWNR 000003     |
| CIRNUM= ***** GX                     | FSLT 000724R     | L\$11R= 000000   | SERFLG= ***** GX | S.PSS 000022      |
| CIRUNT= ***** GX                     | F\$SLVL= 000001  | L.COSI 000015    | SERHST= ***** GX | S.SEC 000040      |
| C\$CKP= 000000                       | F.NXT 000000     | L.CTL 000012     | SERINI 000056RG  | S.TER 000042      |
| C\$SORE= 000400                      | F.USE 000002     | L.CVA 177776     | SERNOD= ***** GX | T\$KMG= 000000    |
| C\$RSH= 177564                       | F.VAR 000003     | L.DDM 000002     | SERPAS= ***** GX | T\$MIN= 000000    |
| DECPT = ***** GX                     | G\$STPP= 000000  | L.DDS 000004     | SERSLN 000000RG  | V\$CTR= 001000    |
| D\$AMXC 000072                       | G\$STSS= 000000  | L.DLC 000003     | SERSLT= ***** GX | X\$DBT= 000000    |
| D\$AMXH 000074                       | G\$STTK= 000000  | L.DLM 000006     | SERTMP= ***** GX | ZF.CDU= 001000    |
| D\$ANN 000000                        | G\$SWRD= 000000  | L.DLS 000010     | SF.ACT= 000200   | ZF.DDM= 000001    |
| D\$BRPR 000102                       | HEXIND 000604R   | L.FLG 000000     | SF.CIR= 000010   | ZF.DIA= 004000    |
| D\$BRIM 000100                       | HEXSTR= ***** GX | L.KRBA 000016    | SF.DEV= 000020   | ZF.DLC= 000002    |
| D\$DEL 000045                        | HRDADD 000074RG  | L.LEN = 000022   | SF.DIA= 004000   | ZF.DVP= 100000    |
| D\$DELW 000046                       | I\$RAR= 000000   | L.MPF 000022     | SF.DPA= 000100   | ZF.INI= 040000    |
| D\$END = 000104                      | I\$RDN= 000000   | L.NMST 000020    | SF.DPC= 000200   | ZF.KMX= 000020    |
| D\$FNB 000034                        | KSARS = ***** GX | L.NSTA 000014    | SF.DUM= 010000   | ZF.LLC= 000004    |
| D\$HIO 000024                        | K\$CNT= 177546   | L.OWNR 000021    | SF.ENA= 000100   | ZF.LMC= 000100    |
| D\$HOST 000022                       | K\$CSR= 177546   | L.UNT 000013     | SF.HAD= 000004   | ZF.LMN= 020000    |
| D\$INAC 000044                       | K\$SLDC= 000000  | M\$CRB= 000124   | SF.HST= 000002   | ZF.MFL= 000010    |
| D\$INCT 000042                       | K\$STPS= 000074  | M\$CRX= 000000   | SF.LDA= 002000   | ZF.MTM= 000400    |
| D\$IPL 000051                        | LD\$LP = 000000  | M\$FCS= 000000   | SF.LPB= 000004   | ZF.MUX= 000040    |
| D\$ID 000020                         | LET = 000100     | M\$MGE= 000000   | SF.MFL= 000040   | ZF.PSE= 002000    |
| D\$INAM 000006                       | LF.ACT= 100000   | M\$NET= 000000   | SF.PAC= 000020   | ZF.SLI= 010000    |
| D\$INUM 000014                       | LF.BRO= 000400   | M\$OVR= 000000   | SF.PH3= 020000   | ZF.TIM= 000200    |
| D\$IST 000047                        | LF.BWT= 000007   | N\$ACC= 000001   | SF.PSS= 000040   | ZF.X3P= 000000    |
| D\$MAXC 000064                       | LF.ENA= 002000   | N\$BUF= 000001   | SF.REA= 000010   | ZS.ASN= 100000    |
| D\$MAXH 000066                       | LF.LPB= 001000   | N\$LDV= 000001   | SF.SEC= 000400   | ZS.BSY= 140000    |
| D\$MAXV 000070                       | LF.MDC= 000100   | N\$MCP= 000001   | SF.SER= 000001   | Z.AVL 000014      |
| D\$MLL 000040                        | LF.MFL= 004000   | N\$MLL= 000001   | SF.SVC= 000002   | Z.DAT 000016      |
| D\$MNOD 000041                       | LF.MTP= 000020   | N\$MOV= 000010   | SF.TER= 001000   | Z.DSP 000000      |
| D\$NA 000062                         | LF.PAC= 000200   | N\$NCT= 000001   | SF.UNL= 000040   | Z.FLG 000010      |
| D\$NBEA 000056                       | LF.RDY= 040000   | N\$PEM= 000001   | SLTMA = ***** GX | Z.LEN = 000016    |
| D\$NBRA 000054                       | LF.REA= 010000   | PAS 000460RG     | SLTMM = ***** GX | Z.LEN 000006      |
| D\$NEND= 000054                      | LF.SER= 000040   | PDVNM = ***** GX | S\$SWRG= 000000  | Z.MAP 000020      |
| D\$NLN 000030                        | LF.TIM= 000010   | PDVTA = ***** GX | S\$YSZ= 007600   | Z.NAM 000004      |
| D\$NN 000060                         | LF.UNL= 020000   | P\$P45= 000000   | S.ADD 000004     | Z.PCB 000012      |
| D\$OUTT 000043                       | LF.X2P= 000000   | P\$SWRD= 000000  | S.CIR 000016     | Z.SCH 000007      |
| D\$RETF 000050                       | LN.CLO= 000000   | Q\$OPT= 000010   | S.COST 000001    | \$CAT5 = ***** GX |
| D\$RNN 000002                        | LN.DUM= 000005   | RTSPC = ***** GX | S.DEV 000020     | \$CEACX= ***** GX |
| D\$RTMR 000076                       | LN.LOA= 000004   | R\$DER= 000000   | S.DIA 000046     | \$ERRSS 000000R   |
| D\$SEG 000036                        | LN.LOO= 000003   | R\$K11= 000001   | S.DPA 000032     | \$HEADR= ***** GX |
| D\$SER 000032                        | LN.OAU= 000003   | R\$SND= 000000   | S.DPC 000036     | \$LINKX= ***** GX |
| D\$SQRL 000052                       | LN.OFF= 000001   | R\$17M= 000000   | S.DUM 000050     | \$XALOC= ***** GX |
| D\$BUG= 177514                       | LN.ON = 000000   | SERADD= ***** GX | S.FLAG 000002    | .PSTCN= ***** GX  |
| D\$ISK= 000000                       | LN.OOP= 000004   | SERALL 000162R   | S.FLG 000000     | .PSTPT= ***** GX  |
| D\$L11= 000001                       | LN.OPE= 000001   |                  |                  |                   |
| . ABS. 177776 000 (RW,I,GBL,ABS,OVR) |                  |                  |                  |                   |
| 001022 001 (RW,I,LCL,REL,CON)        |                  |                  |                  |                   |
| DATA 000052 002 (RW,I,LCL,REL,CON)   |                  |                  |                  |                   |



```

134      MOV      #SERRXB,2(SP)      ; * SAY PARTITION IS BUSY ???
135
136      .IFF
137
138      000132 032761 000000G 000000G      BIT      #PS.COM,P.STAT(R1)      ; * IS IT A COMMON ?
139      000140 001004      BNE      20$      ; IF NE, YES
140      000142 012766 000000G 000002      MOV      #SERRXA,2(SP)      ; * ERROR - PARTITION NOT COMMON
141      000150 000424      BR      50$      ; *
142      000152 016100 000000G      20$:      MOV      P.MAIN(R1),RO      ; * GET MAIN PCB ADDRESS
143      000156 105761 000001G      TSTB     P.BUSY+1(R1)      ; * MAKE SURE NEITHER IS BUSY
144      000162 001003      BNE      30$      ; *
145      000164 105760 000001G      TSTB     P.BUSY+1(RO)      ; *
146      000170 001404      BEQ      40$      ; *
147      000172 012766 000000G 000002 30$:      MOV      #SERRXB,2(SP)      ; * ERROR - PARTITION BUSY
148      000200 000410      BR      50$      ; *
149      000202 156161 000000G 000001G 40$:      BISB     P.BUSY(R1),P.BUSY+1(R1)      ; * MARK PARTITIONS BUSY
150      000210 156160 000000G 000001G      BISB     P.BUSY(R1),P.BUSY+1(RO)      ; *
151      000216 010167 000000G      MOV      R1,..PCB      ; * SAVE PCB ADDRESS
152
153      .ENDC
154
155      000222      50$:      RETURN      ; * BACK TO USER MODE
156      000224 005700      60$:      TST      RO      ; DID ERROR OCCUR IN KERNEL MODE ?
157      000226 001401      BEQ      70$      ; IF EQ, NO
158      000230      CALL      @ (RO)+      ; GO PRINT ERROR MESSAGE
159      000232 016700 000000G      70$:      MOV      .PCB,RO      ; WAS A PARTITION FOUND ?
160
161      .IF DF M$$MGE
162      000236 001411      BEQ      100$      ; IF EQ, NO - JUST EXIT
163      .IFF
164      BNE      80$      ; IF NE, YES
165      MSG$R      XC      ; ERROR - PARTITION NOT IN SYSTEM
166      .ENDC
167
168      000240 016746 000000G      80$:      MOV      .NTPSZ,-(SP)      ; SAVE RDB/SDB SIZE
169      000244 005216      INC      (SP)      ; PLUS 1 FOR NETLDR HOME BLOCK
170      000246 026026 000000G      CMP      P.BKLS(RO),(SP)+      ; IS PARTITION BIG ENOUGH ?
171      000252 101003      BHI      90$      ; IF HI, YES
172      000254      MSG$R      XD      ; ERROR - PARTITION TOO SMALL
173      000262      90$:
174      .IF NDF M$$MGE
175      MOV      P.REL(RO),R1      ; POINT AT START OF PARTITION
176      MOV      R1,..HOME      ; SAVE HOME BLOCK ADDRESS
177      MOV      RO,(R1)+      ; STORE NTPool PCB ADDRESS
178      CLR      (R1)+      ; NO UMR BLOCK ADDRESS
179      .ENDC
180      .ENDC
181
182      000262      100$:      RETURN
183
184      ;
185      ; SECOND PARTITION NAME (PAR$DF)
186      ;
187
188      000264 012702 000000G      P.PAR2::MOV      #.PNAM2,R2      ; POINT AT WORK AREA
189      000270 010246      PNAME::MOV      R2,-(SP)      ; SAVE POINTER
190      000272 010246      MOV      R2,-(SP)      ; TWICE

```

CF2AC9 CREATED BY MACRO ON 29-JUN-85 AT 00:13 PAGE 1 J 14  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL | VALUE      | REFERENCES                      |
|--------|------------|---------------------------------|
| BRADF  | 000000 R   | 7-84                            |
| BRDJ   | 001056 RG  | #9-389                          |
| CFGBF  | = ***** GX | 7-95                            |
| CFGSZ  | = ***** GX | 7-94                            |
| CIRMUL | = ***** GX | *9-337                          |
| CIRNUM | = ***** GX | *9-334                          |
| CIRUNT | = ***** GX | *9-340                          |
| DADDR  | 001000 RG  | #9-367                          |
| DCT    | 001016 RG  | #9-372                          |
| DECDEF | 000174 R   | 7-90                            |
| ECLLLN | = ***** GX | *9-383                          |
| ECLMC  | 001046 RG  | #9-386                          |
| ECLML  | 001036 RG  | #9-383                          |
| ECLNOD | = ***** GX | *9-386                          |
| EXT    | 000550 RG  | #8-297                          |
| EXTN   | = 000003   | #6-71 8-297                     |
| FE.EXT | = ***** GX | 7-231                           |
| FILBUF | = ***** GX | 8-274 8-301 8-302 *8-306 *8-307 |
| FILE   | 000510 RG  | #8-274                          |
| FMASK  | = ***** GX | 7-231                           |
| FNBUFF | = ***** GX | 8-281 8-312                     |
| F.NXT  | 000000     | #6-63                           |
| F.USE  | 000002     | #6-63                           |
| F.VAR  | 000003     | #6-63                           |
| HSAD   | 000752 RG  | #9-358                          |
| HSAR   | 000732 RG  | #9-352                          |
| ISSAS  | = *****    | 6-65                            |
| LF.ACT | = 100000   | #6-66                           |
| LF.BRO | = 000400   | #6-66                           |
| LF.BWT | = 000007   | #6-66                           |
| LF.ENA | = 002000   | #6-66                           |
| LF.LPB | = 001000   | #6-66                           |
| LF.MDC | = 000100   | #6-66                           |
| LF.MFL | = 004000   | #6-66                           |
| LF.MTP | = 000020   | #6-66                           |
| LF.PAC | = 000200   | #6-66                           |
| LF.RDY | = 040000   | #6-66                           |
| LF.REA | = 010000   | #6-66                           |
| LF.SER | = 000040   | #6-66                           |
| LF.TIM | = 000010   | #6-66                           |
| LF.UNL | = 020000   | #6-66                           |
| LF.X2P | = 000000   | #6-66                           |
| LLCKW  | 000000 RG  | 7-93 #10-398                    |
| LLCST  | 000000 RG  | #10-398                         |
| LN.CLO | = 000000   | #6-66                           |
| LN.DUM | = 000005   | #6-66                           |
| LN.LOA | = 000004   | #6-66                           |
| LN.LOO | = 000003   | #6-66                           |
| LN.OAU | = 000003   | #6-66                           |
| LN.OFF | = 000001   | #6-66                           |
| LN.ON  | = 000000   | #6-66                           |
| LN.OOP | = 000004   | #6-66                           |

```

173
174
175
176
177 000462
178
179 000462
180 000462
181 000462
182
183 000462
184 000462
185
186 000462
187 000462
188
189 000462
190 000462
191
192 000462
193 000462
194
195 000462
196 000462
197
198 000462
199 000462
200
201 000462
202 000462
203
204 000462
205 000462
206
207 000462
208 000462
209
210 000462
211 000462
212
213 000462
214 000462
215
216 000462
217 000462
218
219 000462
220 000462
221
222 000462
223 000462
224
225 000462
226 000462
227
228 000462
229 000462

```

```

;+ DEC$DF MACRO STATE TABLE
;-
  ISTAT$  DECST,DECKW

  STATES$  DECDF
  TRANS$  %DEC$DF%,DECDF1,,1,SYNERR
  TRANS$  $LAMDA,$EXIT

  STATES$  DECDF1
  TRANS$  $NUMBR,,ECLLNK           ; MAX LOGICAL LINKS

  STATES$
  TRANS$  <','>

  STATES$
  TRANS$  $NUMBR,,NODCT           ; MAX NODE COUNTERS

  STATES$
  TRANS$  <','>

  STATES$
  TRANS$  $NUMBR,,INCT           ; INCOMING TIMER

  STATES$
  TRANS$  <','>

  STATES$
  TRANS$  $NUMBR,,OUTT           ; OUTGOING TIMER

  STATES$
  TRANS$  <','>

  STATES$
  TRANS$  $NUMBR,,INAC           ; INACTIVITY TIMER

  STATES$
  TRANS$  <','>

  STATES$
  TRANS$  $NUMBR,,DELF           ; DELAY FACTOR

  STATES$
  TRANS$  <','>

  STATES$
  TRANS$  $NUMBR,,DELW           ; DELAT WEIGHT

  STATES$
  TRANS$  <','>

  STATES$
  TRANS$  $NUMBR,,IPL           ; INPUT PACKET LIMITER

  STATES$
  TRANS$  <','>

```

```

255                                     ASLB R4 ; CALCULATE HIGH BYTE OF LOW 16 BITS
256                                     ASL R4 ; (LOSING BIT 200 FIRST)
257                                     ASL R4 ;
258                                     SWAB R4 ; FORM LOW SIXTEEN BITS OF ADDRESS
259                                     MOV R4,M.UMVL-2(R0) ; SET UP LOW 16 BITS OF ADDRESS
260                                     MOVB R5,M.UMVH-2(R0) ; SET BITS 4 & 5
261                                     DEC R0 ; POINT BACK TO LINK WORD PRESERVING C-BIT
262                                     DEC R0 ;
263                                     RETURN ; RETURN TO CALLER
264                                     .ENDC
265
266                                     000001
267                                     .END

```

CFGX2P      CREATED BY    MACRO    ON 29-JUN-85 AT 00:08      PAGE 4      K 1  
SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES   |
|--------|------------|--|
|        |            | 11-297   |
| .PNUMH | = ***** GX | 11-210    11-220    11-243    11-258    11-269    11-281    11-295 |
| .PSTCN | = ***** GX | 11-201   |
| .PSTPT | = ***** GX | 11-203   |
| .TPARS | = ***** GX | 9-122  |

CFGX3  
Symbol

ASSCH  
ASSCF  
ASSPF  
ASSTF  
BLKSM  
BLKSM  
CERR  
CFGBF  
CFGS2  
CFLIN  
CHKBL  
CHKWN  
C\$\$\$CH  
C\$\$\$OF  
C\$\$\$RS  
D\$\$\$BU  
D\$\$\$I  
D\$\$\$L  
D\$\$\$Y  
D\$\$\$Y  
E\$\$\$X  
FMT8  
FMT8  
FM.8  
FM.8

. AB  
DATA  
SSTA  
SKTAL  
SKSTI  
Error

\*\*\*

Work  
Work  
Size  
Size  
Open

Elap  
SY:C

|                  |                 |                  |                     |                       |     |
|------------------|-----------------|------------------|---------------------|-----------------------|-----|
| AS\$CHK= 000000  | F\$SLVL= 000001 | N\$SLDV= 000001  | X3DEV 000002R       | 002 \$ERR2E 000122R   | 002 |
| AS\$CPS= 000000  | G\$STPP= 000000 | N\$SMCP= 000001  | X3NXT 000042R       | 003 \$ERR2F= ***** GX |     |
| AS\$PRI= 000000  | G\$STSS= 000000 | N\$SMLL= 000001  | X3PDF 000000R       | 003 \$ERR2G 000156R   | 002 |
| AS\$TRP= 000000  | G\$STTK= 000000 | N\$SMOV= 000010  | X3PKW 000000RG      | 004 \$ERR2H 000232R   | 002 |
| BLKSMN= 000040   | G\$SWRD= 000000 | N\$SNCT= 000001  | X3PST 000000RG      | 003 \$EXIT = 000000   |     |
| BLKSMX= 002000   | I\$SRAR= 000000 | N\$SPEM= 000001  | X3TMCT= 000007      | \$FAIL = 177777       |     |
| CERR = ***** GX  | I\$SRDN= 000000 | P\$SP45= 000000  | X3UNT 000006R       | 002 \$LAMD= 000000    |     |
| CFGBF = ***** GX | K\$SCNT= 177546 | P\$SWRD= 000000  | X3.BEG 000212R      | \$NUMB= 000002        |     |
| CFGSZ = ***** GX | K\$SCSR= 177546 | Q\$SOPT= 000010  | X3.DBS 000052R      | \$QX3P 000000RG       |     |
| CFLIN = ***** GX | K\$SLDC= 000000 | RTSPC = ***** GX | X3.DWS 000132R      | \$RAD50= 000016       |     |
| CHKBLK= ***** GX | K\$STPS= 000074 | R\$SDER= 000000  | X3.END 000312R      | \$STRNG= 000004       |     |
| CHKWND= ***** GX | LD\$LP = 000000 | R\$SK11= 000001  | X3.MBS 000074R      | \$SUBXP= 000010       |     |
| C\$CKP= 000000   | L\$ASG= 000000  | R\$SND= 000000   | X3.MWS 000154R      | \$X3CAL= ***** GX     |     |
| C\$SORE= 000400  | L\$DRY= 000000  | R\$S11M= 000000  | X3.NXT 000224R      | \$X3DBS= ***** GX     |     |
| C\$RSH= 177564   | L\$P11= 000001  | SYNERR= ***** GX | \$ALPHA= 000022     | \$X3DWS= ***** GX     |     |
| D\$BUG= 177514   | L\$P11R= 000000 | S\$SWRG= 000000  | \$ANY = 000020      | \$X3MBS= ***** GX     |     |
| D\$ISK= 000000   | M\$CRB= 000124  | S\$SYZ= 007600   | \$BLANK= 000006     | \$X3MWS= ***** GX     |     |
| D\$LL1= 000001   | M\$CRX= 000000  | TIMRCT 000010R   | \$DIGIT= 000024     | \$SFLG= 177777        |     |
| D\$SYNC= 000000  | M\$FCS= 000000  | T\$KMG= 000000   | \$DNUMB= 000014     | \$SKEY= 000000        |     |
| D\$SYNM= 000000  | M\$MGE= 000000  | T\$MIN= 000000   | \$EOS = 000012      | \$SSTA= 000042R       | 003 |
| E\$XPR= 000000   | M\$NET= 000000  | V\$CTR= 001000   | \$ERR1T= ***** GX   | \$STMP= 000000R       | 005 |
| FMT8 = ***** GX  | M\$OVR= 000000  | WNSMX= 000177    | \$ERR2B 000012R     | .PNUMB= ***** GX      |     |
| FMT8B = ***** GX | NEXT 000000R    | X\$SDBT= 000000  | \$ERR2C= ***** GX   | .PNUMH= ***** GX      |     |
| FM.8 = 000000    | N\$ACC= 000001  | X3CTL 000004R    | 002 \$ERR2D 000046R | 002 .TPARS= ***** GX  |     |
| FM.8B = 000000   | N\$BUF= 000001  |                  |                     |                       |     |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
 000330 001 (RW,I,LCL,REL,CON)  
 DATA 000256 002 (RW,D,LCL,REL,CON)  
 \$STATE 000060 003 (RW,D,LCL,REL,CON)  
 \$KIAB 000002 004 (RW,D,LCL,REL,CON)  
 \$KSTR 000007 005 (RW,D,LCL,REL,CON)  
 Errors detected: 0

### \*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 12456 Words ( 49 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:26.17  
 SY:CFGX3P.V2,132,134]C^GX3P/CR/-SP=SY:[1,1]RSXMCM.SML/ML,130,110]NETLIB/ML,130,10]RSXMCM/PA:1,132,10]CFGX3P

```

235      ; REM$DF ACTION ROUTINES
236
237
238      000312 012700 000000' R.NAME::MOV #NAME,RO ; PLACE TO STORE NODE NAME
239      000316 012710 020040      MOV #'',(RO) ; PAD NODE NAME WITH BLANKS
240      000322 012010      MOV (RO)+,(RO) ; ...
241      000324 012010      MOV (RO)+,(RO) ; ...
242      000326 022767 000006 000000G      CMP #6,PSTCN ; IS THE NODE NAME THE RIGHT LENGTH ?
243      000334 103411      BLO 101$ ; IF LO, NO
244      000336 012700 000000'      MOV #NAME,RO ; STORE THE NODE NAME
245      000342 016701 000000G      MDV .PSTPT,R1 ; ...
246      000346 112120      10$: MDVB (R1)+,(RO)+ ; ...
247      000350 005367 000000G      DEC .PSTCN ; LDOP <.PSTCN> TIMES
248      000354 003374      BGT 10$ ; ...
249      000356      RETURN
250
251      ; ERROR CONDITIONS
252
253      000360      101$: MSG$R YX ; ILLEGAL NODE NAME
254
255
256      000366 022767 000077 000000G R.ARE::CMP #63,..PNUMB ; CHECK HIGH LIMIT
257      000374 002420      BLT 101$ ; IF ERROR - BRANCH
258      000376 016700 000000G      MOV .PNUMB,RO ; STORE VALUE IN A REGISTER
259      000402      ASL$ 10,.RO ; POSITION AREA BITS
260      000426 010067 000006'      MOV RO,RADDR ; STORE AREA
261      000432 001401      BEQ 101$ ; IF ERROR - BRANCH
262      000434      RETURN
263
264      000436      101$: MSG$R YZ ; ILLEGAL AREA ADDRESS
265
266
267      000444 022767 001777 000000G R.NUM::CMP #1023,..PNUMB ; CHECK FOR HIGH LIMIT
268      000452 002512      BLT 101$ ; IF ERROR - BRANCH
269      000454 036767 000000G 000006'      BIS .PNUMB,RADDR ; SAVE REMOTE ADDRESS
270      000462 012701 000014      MOV #R.LEN,R1 ; GET THE LENGTH OF A REMOTE BLOCK
271      000466      SWSTK$ 30$ ; * ENTER KERNEL MODE
272      000472      CALL $XALOC ; * TRY TO ALLOCATE NODE NAME BLOCK
273      000476 103006      BCC 10$ ; * BR IF SUCCESS
274      000500      RETC RO ; * SET USER C-BIT
275      000512 000467      BR 20$ ; * AND EXIT
276      000514      10$: SAVMAP ; * SAVE CURRENT MAPPING
277      000520 010002      MOV RO,R2 ; * SAVE BUFFER ADDRESS
278      000522 010046      MOV RO,-(SP) ; * SET BUFFER ADDRESS
279      000524      CALL $CEACX ; * GET ACCESS TO BUFFER
280      000530 012600      MOV (SP)+,RO ; * GET MAPPED BUFFER ADDRESS
281      000532 016760 000000' 000002      MOV NAME,R.NAM(RO) ; * STORE THE NODE NAME
282      000540 016760 000002' 000004      MOV NAME+2,R.NAM+2(RO) ; * ...
283      000546 016760 000004' 000006      MOV NAME+4,R.NAM+4(RO) ; * ...
284      000554 016760 000006' 000010      MOV RADDR,R.ADD(RO) ; * STORE THE NODE ADDRESS
285      000562 005060 000012      CLR R.FLAG(RO) ; * INITIALIZE FLAGS WORD
286
287      000566      CALL QUICK ; * ATTEMPT FAST REMOTE LIST SETUP
288      000572 103031      BCC 18$ ; * IF SUCCESS - BRANCH
289      000574 000241      CLC ; * IF FAILURE - BEGIN AT HEAD OF LIST
290
291      000576 017700 000000G      MOV @DECPT,RO ; * GET ADDRESS OF DECNET NAME BLOCK
  
```

```
91 .SBTTL ERROR MESSAGES
92
93
94 ; ERROR MESSAGES
95 ;
96 .ENABL LC
97 .NLIST BEX
98
99 000010 NTLR$ ,XW,8,CERR,RTSPC,CFLIN,<Illegal object number>
100 000046 NTLR$ ,XX,8,CERR,RTSPC,CFLIN,<Illegal object task name>
101 000110 NTLR$ ,XY,8,CERR,RTSPC,CFLIN,<Illegal object verification>
102 000154 NTLR$ ,XZ,8,CERR,RTSPC,CFLIN,<Illegal object type flags word>
103 000224 NTLR$ ,YY,8,CERR,RTSPC,<Object or Remote block allocation failure>
104 000306 NTLR$ ,YZ,8,CERR,RTSPC,CFLIN,<Too many object task copies>
105
106 .DSABL LC
107 .NLIST BEX
108 .EVEN
109 000000 .PSECT
```



```

180 000202 006001          ROR      R1          ;
181 000204          SOB      R2,10$          ;
182          ;IF
183          BEQ      20$          ; IF NO OVERFLOW, OK
184          MOV      #-1,R1          ; ELSE CAUSE ALLOCATION TO FAIL
185          BIC      #77,R1          ; LEAVING ALLOCATION SIZE
186          .ENDC
187 000210 010167 000000G    MOV      R1,.NTPSZ          ; SAVE RDB/SDB SIZE REQUIREMENTS
188
189 000214          RETURN
190
191 000216          ;101$:  MSG$R  XS          ; ERROR - TOO FEW ARGS

```

```

54          .SBTTL  MACRO CALLS
55
56      ;+
57      ; LIBRARY MACRO CALLS
58      ;
59      ; -
60
61          .MCALL  FLTDF$,EMSG$R,RETC,NTLR$
62
63      000000      FLTDF$          ; DEFINE EVENT FILTER CONTROL BLOCK OFFSETS
64
65      ; LOCAL MACRO DEFINITIONS
66      ;
67      ;
68      .MACRO  SAVMAP
69      MOV     KSAR5,-(SP)      ; SAVE APR 5
70      .ENDM
71
72      .MACRO  RESMAP
73      MOV     (SP)+,KSAR5      ; RESTORE APR5
74      .ENDM
75

```

```

61
62
63      ;+
64      ; OFFSET VALUES
65      ; -
66 000000      DHBDF$      ; DEFINE DECNET HOME BLOCK OFFSETS
67 000000      FNBD$      ; DEFINE FILE NAME BLOCK OFFSETS
68 000000      SERDF$      ; DEFINE SERVICE BLOCK OFFSETS
69 000000      PDVDF$      ; DEFINE PDV OFFSETS
70 000000      SLTDF$      ; DEFINE SLT OFFSETS
71
72
73
74      ;+
75      ; LOCAL ERROR MESSAGES
76      ; -
77
78 000000      .PSECT DATA
79
80 000000      NTLERS$ ,FS,8,CERR,REP8C,CFLIN,<File block allocation failure>
81
82      .EVEN
83      .PSECT
84
85

```

CF2AC5      CREATED BY    MACRO    ON 29-JUN-85 AT 00:11      PAGE 2      K 8

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|---|
| L.COST  | 000015     | #6-70   |
| L.CTL   | 000012     | #6-70   |
| L.CVA   | 177776     | #6-70   |
| L.DDM   | 000002     | #6-70   |
| L.DDS   | 000004     | #6-70   |
| L.DLC   | 000003     | #6-70   |
| L.DLM   | 000006     | #6-70   |
| L.DLS   | 000010     | #6-70   |
| L.FLG   | 000000     | #6-70   |
| L.KRBA  | 000016     | #6-70   |
| L.LEN   | = 000022   | #6-70   |
| L.MPF   | 000022     | #6-70   |
| L.NMST  | 000020     | #6-70   |
| L.NSTA  | 000014     | #6-70   |
| L.OWNR  | 000021     | #6-70   |
| L.UNIT  | 000013     | #6-70   |
| MOVBUF  | 000174 R   | #9-204      10-243      10-247      10-252                              |
| NS\$VCT | = *****    | 7-109      11-278   |
| REP8C   | = ***** GX | 6-80  |
| RS\$11D | = *****    | 6-69  |
| RS\$11M | = 000000   | 6-69  |
| RS\$11S | = *****    | 6-69  |
| SERCUR  | = ***** GX | 12-373  |
| SERFLG  | = ***** GX | *12-338      *12-342      *12-346      *12-350      *12-354      12-376 |
| SEROFF  | = ***** GX | *12-337      *12-341      *12-345      *12-349      *12-353      12-377 |
| SETBLK  | 000216 R   | #10-230      12-356   |
| SETST   | 000720 RG  | 12-366      #12-385   |
| SF.ACT  | = 000200   | #6-70   |
| SF.CIR  | = 000010   | #6-68   |
| SF.DEV  | = 000020   | #6-68   |
| SF.DIA  | = 004000   | #6-68      12-350   |
| SF.DPA  | = 000100   | #6-68   |
| SF.DPC  | = 000200   | #6-68   |
| SF.DUM  | = 010000   | #6-68      12-354   |
| SF.ENA  | = 000100   | #6-70   |
| SF.HAD  | = 000004   | #6-68   |
| SF.HST  | = 000002   | #6-68   |
| SF.LOA  | = 002000   | #6-68      12-346   |
| SF.LPB  | = 000004   | #6-70   |
| SF.MFL  | = 000040   | #6-70   |
| SF.PAC  | = 000020   | #6-70   |
| SF.PH3  | = 020000   | #6-68   |
| SF.PSS  | = 000040   | #6-68   |
| SF.REA  | = 000010   | #6-70   |
| SF.SEC  | = 000400   | #6-68      12-338   |
| SF.SER  | = 000001   | #6-70   |
| SF.SVC  | = 000002   | #6-70   |
| SF.TER  | = 001000   | #6-68      12-342   |
| SF.UNL  | = 000040   | #6-70   |
| SRHFNB  | 000000 R   | #7-101      12-357  |
| STRPT   | = ***** GX | *12-385   |
| STRSZ   | = ***** GX | *12-386   |

```

274                                     ; PROCESS NAME (LLC$DF, DLC$DF, DDM$DF)
275
276
277 000560 016700 000000G PRONM:: MOV    ,PSTPT,RO ; GET TPARS POINTER
278 000564 010701          MOV    PC,R1 ; ALLOW ... AS CHARACTER
279 000566          CALL    $CAT5 ; CONVERT TO RAD50
280 000572 010167 000000' MOV    R1,NAME ; SAVE THE NAME
281 000576 017700 000000G MOV    @PDVTA,RO ; GET PDV TABLE ADDRESS
282
283 000602 017701 000000G MOV    @PDVNM,R1 ; GET PDV COUNT
284 000606 001410          BEQ    40$ ; IF ZERO, ERROR
285 000610 012002          10$: MOV    (R0)+,R2 ; GET PDV ADDRESS
286 000612 001404          BEQ    30$ ; IF ZERO, SKIP THIS SLOT
287
288 000614 026267 000000G 000000' 20$: CMP    Z,NAM(R2),NAME ; DO NAMES MATCH ?
289 000622 001405          BEQ    50$ ; IF EQ, YES
290 000624          30$: SOB    R1,10$ ; LOOP <PDV COUNT> TIMES
291 000630          40$: EMMSG$R XN ; ERROR - PROCESS NOT IN SYSTEM
292 000636 010267 000000G 50$: MOV    R2,$PDVA ; SAVE PDV ADDRESS
293 000642          RETURN
294
295
296                                     ; PROCESS MARKED-FOR-LOAD FLAG (LLC$DF, DLC$DF, DDM$DF)
297
298 000644 016700 000000G PRMFL:: MOV    $PDVA,RO ; GET PDV ADDRESS
299 000650 152760 000000G 000000G BISB    #ZF.MFL,Z.FLG(R0) ; SET MARKED-FOR-LOAD FLAG
300 000656          10$: RETURN
  
```

```

74
75
76
77 000000 000000          K6TMP: .WORD 0          ; $K6TAB TEMP LOCATION
78
79
80
81
82
83
84
85
86
87 000002 016767 000000G 000000G P.ALL:: MOV .PNUMB,.ALLO  ; STORE LENGTH OF DSR EXTENSION
88 000010 066767 000000G 000000G ADD .ALLO,.FUDGE ; UPDATE PARTITION SIZE
89
90 000016 016700 000000G P.END:: MOV .NN,R0 ; GET MAX NODES
91 000022 001471 BEQ 1$ ; IF ENDNODE - BRANCH
92 000024 066700 000000G ADD .NA,R0 ; INCLUDE MAX AREAS
93 000030 062700 000004 ASL #4,R0 ; ALLOCATE FOR XPT ROL, MIN HOP/COST OFFSET 0
94 000034 006300 R0 ; CONVERT TO BYTES
95 000036 062700 000077 ADD #77,R0 ; ROUND UP TO NEXT BLOCK
96 000042 006300 ASL R0 ; ALLOCATE MIN HOP/COST MATRIX
97 000044 062700 000077 ADD #77,R0 ; ROUND UP TO NEXT BLOCK
98 000050 042700 000077 BIC #77,R0 ; ...
99 000054 000241 CLC
100 000056 006200 ASR R0 ; CONVERT TO BLOCKS
101 000060 006200 ASR R0 ; ...
102 000062 006200 ASR R0 ; ...
103 000064 006200 ASR R0 ; ...
104 000066 006200 ASR R0 ; ...
105 000070 006200 ASR R0 ; ...
106 000072 060067 000000G ADD R0,.NOMAP ; UPDATE NON UMR MAPPED POOL
107 000076 060067 000000G ADD R0,.FUDGE ; UPDATE PARTITION SIZE
108 000102 005767 000000G TST .NN ; ENDNODE ?
109 000106 001437 BEQ 1$ ; IF YES - BRANCH
110 000110 005002 CLR R2 ; CLEAR FLAG
111 000112
112 000112 016700 000000G 3$: MOV XPTLLN,R0 ; STORE XPT LINES
113 000116 005200 INC R0 ; INCLUDE LOOPBACK CHANNEL
114 000120 066700 000000G ADD .NBEA,R0 ; INCLUDE BROADCAST ENDNODE ADJACENCIES
115 000124 066700 000000G ADD .BRAD,R0 ; INCLUDE ROUTER ADJACENCIES
116 000130 012701 000004 MOV #ASLEN,R1 ; COPY ADJACENCY DATA BASE SIZE
117 000134 CALL $MUL ; CALCULATE TOTAL SIZE
118 000140 062701 000077 ADD #77,R1 ; CONVERT TO BLOCKS
119 000144 042701 000077 BIC #77,R1 ; ...
120 000150 000241 CLC
121 000152 006201 ASR R1 ; ...
122 000154 006201 ASR R1 ; ...
123 000156 006201 ASR R1 ; ...
124 000160 006201 ASR R1 ; ...
125 000162 006201 ASR R1 ; ...
126 000164 006201 ASR R1 ; ...
127 000166 060167 000000G ADD R1,.NOMAP ; ADJUST TOTAL OF NON UMR BLOCK POOL
128 000172 060167 000000G ADD R1,.FUDGE ; ADJUST TOTAL POOL
129 000176 005702 TST R2 ; SECOND BLOCK ALLOCATED ?
130 000200 001002 BNE 1$ ; IF YES - BRANCH

```

\*\*\*FILE\*\*ID\*\*CF2AC8

K 11

```
CCCCCCCC  FFFFFFFF  222222  AAAAAA  CCCCCC  888888
CCCCCCCC  FFFFFFFF  222222  AAAAAA  CCCCCC  888888
CC         FF         22         AA         CC         88         88
CC         FF         22         AA         CC         88         88
CC         FF         22         AA         CC         88         88
CC         FF         22         AA         CC         88         88
CC         FFFFFFFF  22         AA         CC         888888
CC         FFFFFFFF  22         AA         CC         888888
CC         FF         22         AA         CC         88         88
CC         FF         22         AA         CC         88         88
CC         FF         22         AA         CC         88         88
CC         FF         22         AA         CC         88         88
CC         FF         22         AA         CC         88         88
CCCCCCCC  FF         2222222222  AA         CCCCCC  888888
CCCCCCCC  FF         2222222222  AA         CCCCCC  888888
```

```
LL         SSSSSSSS  TTTTTTTTTT
LL         SSSSSSSS  TTTTTTTTTT
LL         SS         TT
LL         SS         TT
LL         SS         TT
LL         SS         TT
LL         SSSSSS     TT
LL         SSSSSS     TT
LL         SS         TT
LL         SS         TT
LL         SS         TT
LL         SS         TT
LL         SS         TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT
```

CF2AC8 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:12 Page 11-2  
Symbol table

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15543 Words ( 61 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:15.55  
SY:CF2AC8.V2,[132,134]CF2AC8/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CF2AC8



```

191 000274 012700 000004G      MOV      #,NAMBF+4,R0      ; FILL LOCAL BUFFER WITH BLANKS
192 000300 012710 020040      MOV      #, (R0)
193 000304 011040      MOV      (R0),-(R0)
194 000306 011040      MOV      (R0),-(R0)
195 000310 016701 000000G      MOV      .PSTPT,R1      ; COPY TPARS STRING INTO BUFFER
196 000314 016702 000000G      MOV      .PSTCN,R2      ; BUT NO MORE THAN SIX CHARACTERS
197 000320 022702 000006      CMP      #6,R2
198 000324 103002      BHS      10$
199 000326 012702 000006      MOV      #6,R2
200 000332 112120      10$:      MOVVB   (R1)+,(R0)+
201 000334      SOB      R2,10$
202 000340 012700 000000G      MOV      #,NAMBF,R0      ; NOW CONVERT FROM ASCII
203 000344 010701      MOV      PC,R1      ; TO RAD50
204 000346      CALL     $CAT5      ; ALLOWING PERIODS
205 000352 010136      MOV      R1,a(SP)+
206 000354 010701      MOV      PC,R1
207 000356      CALL     $CAT5
208 000362 012602      MOV      (SP)+,R2
209 000364 010162 000002      MOV      R1,2(R2)
210 000370      RETURN
211
212      ;
213      ; FUDGE FACTOR (PAR$DF)
214      ;
215      ;
216 000372 016767 000000G 000000G P.FUDG: MOV      M$MGE      ; SAVE THE NUMBER
217 000400 001002      BNE      10$      ; IF NE, OK
218 000402 005267 000000G      INC      .FUDGE      ; ELSE CHANGE 0 INTO 1
219 000406      10$:      RETURN
220      .ENDC
221
222      ;
223      ; SIZE OF NON-UMR-MAPPED AREA (PAR$DF)
224      ;
225      ;
226 000410 016767 000000G 000000G P.UMR:  MOV      M$MGE      ; COPY THE SIZE
227 000416 001002      BNE      10$      ; IF NE, OK
228 000420 005267 000000G      INC      .NOMAP      ; ELSE CHANGE 0 INTO 1
229 000424      10$:
230      ;
231 000424 032777 000000G 000000G      IF DF      R$11M
232 000432 001423      BIT      #FE,EXT,aFMASK      ; IS THIS AN 11/70 ?
233      BEQ      P.NONE      ; IF EQ, NO - DISABLE UMR'S
234      .ENDC
235 000434 016700 000000G      MOV      .NTPSZ,R0      ; GET RDB/SDB SIZE
236 000440 066700 000000G      ADD      .FUDGE,R0      ; PLUS FUDGE FACTOR
237
238 000444 016701 000000G      IF DF      R$11M
239 000450 001402      MOV      .PCB,R1      ; GET NTPool PCB ADDRESS
240 000452 016100 000000G      BEQ      20$      ; IF EQ, NO PCB - USE BUF$DF NUMBERS
241      MOV      P.REL(R1),R0      ; IF PCB, USE ITS SIZE
242      .IFF
243      MOV      .REG,R1      ; GET GCD NODE ADDRESS FOR NTPool
244      BEQ      20$      ; IF EQ, NO GCD - USE BUF$DF NUMBERS
245      MOV      G.CZ(R1),R0      ; IF GCD, USE ITS SIZE
246      .ENDC
247 000456 166700 000000G      20$:      SUB      .NOMAP,R0      ; LESS SIZE OF NON-UMR-MAPPED AREA

```

CF2AC9 CREATED BY MACRO ON 29-JUN-85 AT 00:13 PAGE 2 K 14

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES                            |
|---------|------------|---------------------------------------|
| LN.OPE  | = 000001   | #6-66                                 |
| LN.REF  | = 000002   | #6-66                                 |
| LN.SER  | = 000002   | #6-66                                 |
| LN.STA  | = 000017   | #6-66                                 |
| LN.SUB  | = 000360   | #6-66                                 |
| LN.TRI  | = 000006   | #6-66                                 |
| L.COST  | 000015     | #6-66                                 |
| L.CTL   | 000012     | #6-66                                 |
| L.CVA   | 177776     | #6-66                                 |
| L.DDM   | 000002     | #6-66                                 |
| L.DDS   | 000004     | #6-66                                 |
| L.DLC   | 000003     | #6-66                                 |
| L.DLM   | 000006     | #6-66                                 |
| L.DLS   | 000010     | #6-66                                 |
| L.FLG   | 000000     | #6-66                                 |
| L.KRBA  | 000016     | #6-66                                 |
| L.LEN   | = 000022   | #6-66                                 |
| L.MPF   | 000022     | #6-66                                 |
| L.NMST  | 000020     | #6-66                                 |
| L.NSTA  | 000014     | #6-66                                 |
| L.DWNR  | 000021     | #6-66                                 |
| L.UNT   | 000013     | #6-66                                 |
| M\$MGE  | = 000000   | 7-161 7-174 7-215 7-225 7-257         |
| NOAD    | 000722 RG  | #9-349                                |
| NOAR    | 000702 RG  | #9-343                                |
| PARHD   | = ***** GX | 7-122                                 |
| PNAME   | 000270 RG  | #7-189 7-118                          |
| PS.COM  | = ***** GX | 7-138                                 |
| P.BLKS  | = ***** GX | 7-170                                 |
| P.BUSY  | = ***** GX | 7-143 7-145 7-149 *7-149 7-150 *7-150 |
| P.FUDG  | 000372 RG  | #7-216                                |
| P.MAIN  | = ***** GX | 7-142                                 |
| P.NAM   | = ***** GX | 7-127 7-129                           |
| P.NONE  | 000502 RG  | #7-258 7-232                          |
| P.PAR1  | 000066 RG  | #7-117                                |
| P.PAR2  | 000264 RG  | #7-188                                |
| P.REL   | = ***** GX | 7-240                                 |
| P.STAT  | = ***** GX | 7-138                                 |
| P.UMR   | 000410 RG  | #7-226                                |
| R\$SMPL | = *****    | 7-132                                 |
| R\$11D  | = *****    | 6-65                                  |
| R\$11M  | = 000000   | 6-65 7-120 7-230 7-237                |
| R\$11S  | = *****    | 6-65                                  |
| SERDCT  | = ***** GX | *9-372                                |
| SERDEV  | = ***** GX | *9-363                                |
| SERDMP  | = ***** GX | *9-367                                |
| SERDV   | 000762 RG  | *9-361                                |
| SERHST  | = ***** GX | *9-355                                |
| SERMUL  | 000662 RG  | *9-337                                |
| SERMUX  | 000672 RG  | *9-340                                |
| SERNOD  | = ***** GX | *9-346                                |
| SERUNI  | 000652 RG  | *9-334                                |

```
230  
231 000462          STATES$  
232 000462          TRANS$ $NUMBR,,LST          ; LINK SERVICE THRESHOLD  
233  
234 000462          STATES$  
235 000462          TRANS$ <','>  
236  
237 000462          STATES$  
238 000462          TRANS$ $NUMBR,,RETF          ; RETRANSMIT FACTOR  
239  
240 000462          STATES$  
241 000462          TRANS$ <','>  
242  
243 000462          STATES$  
244 000462          TRANS$ $NUMBR,$EXIT,SEG      , ECL SEGMENT SIZE  
245  
246 000462          STATES$  
247  
248  
249          000001          .END
```

Symbol table

|                    |                |                  |                  |                   |
|--------------------|----------------|------------------|------------------|-------------------|
| ALOCB = ***** GX   | G\$TSS= 000000 | M\$FCS= 000000   | PR1 = 000040     | TPS = 177564      |
| A\$CHK= 000000     | G\$TTK= 000000 | M\$MGE= 000000   | PR2 = 000100     | T\$KMG= 000000    |
| A\$CPS= 000000     | G\$WRD= 000000 | M\$NET= 000000   | PR3 = 000140     | T\$MIN= 000000    |
| A\$PRI= 000000     | I\$RAR= 000000 | M\$OVR= 000000   | PR4 = 000200     | UBMPR= 170200     |
| A\$TRP= 000000     | I\$RDN= 000000 | M.BFVH= ***** GX | PR5 = 000240     | UISARO= 177640    |
| BUFUMP= 172354     | KISAR= 172340  | M.BFVL= ***** GX | PR6 = 000300     | UISAR1= 177642    |
| CMODE = 140000     | KISAR6= 172354 | M.LGTH= ***** GX | PR7 = 000340     | UMRLD 000244R     |
| C\$CKP= 000000     | K\$CNT= 177546 | M.UMRA= ***** GX | PS = 177776      | V\$CTR= 001000    |
| C\$ORE= 000400     | K\$CSR= 177546 | M.UMRN= ***** GX | P\$P45= 000000   | X\$DBT= 000000    |
| C\$RSH= 177544     | K\$LDC= 000000 | N\$ACC= 000001   | P\$WRD= 000000   | \$ASUMR= ***** GX |
| DEACB = ***** GX   | K\$TPS= 000074 | N\$BUF= 000001   | P.BLKS= ***** GX | \$ERRXL= ***** GX |
| D\$BUG= 177514     | LD\$LP= 000000 | N\$LDV= 000001   | P.REL = ***** GX | \$ERRXM= ***** GX |
| D\$ISK= 000000     | L\$ASG= 000000 | N\$MCP= 000001   | Q\$OPT= 000010   | \$SAVRG= ***** GX |
| D\$L11= 000001     | L\$DRV= 000000 | N\$MLL= 000001   | R\$DER= 000000   | \$UMRAL 000000R6  |
| D\$YNC= 000000     | L\$P11= 000001 | N\$MOV= 000010   | R\$K11= 000001   | .ALLOc= ***** GX  |
| D\$YNM= 000000     | L\$11R= 000000 | N\$NCT= 000001   | R\$SND= 000000   | .BASEB= 140000    |
| E\$XPR= 000000     | MPAR = 172100  | N\$PEM= 000001   | R\$11M= 000000   | .NOMAP= ***** GX  |
| FE,EXT= ***** GX   | MPCSR = 177746 | PIRQ = 177772    | SWR = 177570     | .PUMR = ***** GX  |
| F\$MASK = ***** GX | M\$CRB= 000124 | PMODE = 030000   | S\$WRG= 000000   | .UMRBA= ***** GX  |
| F\$LVL= 000001     | M\$CRX= 000000 | PRO = 000000     | S\$YSZ= 007600   | .UMRFL= ***** GX  |
| G\$TPP= 000000     |                |                  |                  |                   |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
 000312 001 (RW,I,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 9837 Words ( 39 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:08.54  
 Sy:CF2UMR.V2,L132,134]CF2UMR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CF2UMR

## MACRO CROSS REFERENCE

CREF 04.00

## MACRO NAME REFERENCES

|         |         |         |         |         |         |         |         |         |         |         |  |  |  |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|--|--|--|
| CALL    | 9-122   | 11-204  |         |         |         |         |         |         |         |         |  |  |  |
| DBGTP\$ | #10-136 | #10-154 | #10-191 |         |         |         |         |         |         |         |  |  |  |
| EMSG\$R | #5-60   | 9-130   | 11-238  | 11-254  | 11-265  | 11-277  | 11-291  | 11-306  |         |         |  |  |  |
| ISTAT\$ | #5-60   | 10-136  |         |         |         |         |         |         |         |         |  |  |  |
| MTRAN\$ | #10-136 |         |         |         |         |         |         |         |         |         |  |  |  |
| NTLR\$  | #5-60   | 8-98    | 8-99    | 8-100   |         |         |         |         |         |         |  |  |  |
| PDVDF\$ | #5-60   | 5-62    |         |         |         |         |         |         |         |         |  |  |  |
| REJ\$   | #5-67   | 11-235  |         |         |         |         |         |         |         |         |  |  |  |
| RETURN  | 9-126   | 11-206  | 11-216  | 11-236  | 11-253  | 11-264  | 11-276  | 11-290  | 11-305  |         |  |  |  |
| SLTDF\$ | #5-60   | 5-63    |         |         |         |         |         |         |         |         |  |  |  |
| STATE\$ | #5-60   | 10-140  | #10-143 | #10-146 | #10-149 | #10-152 | #10-156 | #10-159 | #10-162 | #10-165 |  |  |  |
|         | #10-168 | #10-171 | #10-174 | #10-177 | #10-180 | #10-183 | #10-186 | #10-189 | #10-193 |         |  |  |  |
| TRANS   | #5-60   | #10-141 | #10-144 | #10-147 | #10-150 | #10-153 | #10-154 | #10-157 | #10-160 | #10-163 |  |  |  |
|         | #10-166 | #10-169 | #10-172 | #10-175 | #10-178 | #10-181 | #10-184 | #10-187 | #10-190 | #10-191 |  |  |  |

BLKSM  
BLKSM  
CERR  
CFGBH  
CFGSL  
CFLIN  
CHKBL  
CHKW  
FMT8  
FMT8  
FM.8  
FM.8  
NEXT  
RTSP  
SYNE  
S\$B

TIMR  
WINDS  
X3CT  
X3DE  
X3PD  
X3PK  
X3PS  
X3TM  
X3UN  
X3.B  
X3.D  
X3.D  
X3.E  
X3.M  
X3.M  
X3.N  
X3.N  
\$ALP  
\$ANY  
\$BLA  
\$DIG  
\$DNU  
\$EOS  
\$ERR  
\$ERR  
\$ERR  
\$ERR  
\$ERR  
\$ERR  
\$ERR  
\$EXI  
\$FAI  
\$GPR  
\$LAM  
\$NUM

CFGX3P      CREATED BY    MACRO    ON 29-JUN-85 AT 00:09      PAGE 1      L 2

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL   | VALUE       | REFERENCES   |
|----------|-------------|--|
| BLKSMN   | = 000040    | #6-67  |
| BLKSMX   | = 002000    | #6-68  |
| CERR     | = ***** GX  | 8-91      8-92      8-93      8-94      8-95                               |
| CFGBF    | = ***** GX  | 9-117  |
| CFGSZ    | = ***** GX  | 9-116  |
| CFLIN    | = ***** GX  | 8-91      8-92      8-93      8-94      8-95                               |
| CHKBLK   | = ***** GX  | 11-181      11-189   |
| CHKWND   | = ***** GX  | 11-200      11-208   |
| FMT8     | = ***** GX  | 8-92      8-94   |
| FMT8B    | = ***** GX  | 8-91      8-93      8-95   |
| FM.8     | = 000000    | #8-92      #8-94   |
| FM.8B    | = 000000    | #8-91      #8-93      #8-95  |
| NEXT     | = 000000 R  | *11-219      11-231      *11-232   |
| RTSPC    | = ***** GX  | 8-91      8-92      8-94      8-95   |
| SYNERR   | = ***** GX  | *9-118      *9-121      *11-234  |
| SSBAS    | = *****     | 8-91      8-92      8-92      8-93      8-93      8-94      8-94      8-95 |
| TIMRCT   | = 000010 R  | #7-82      *11-220      *11-224      11-225      11-241                    |
| WNSMX    | = 000177    | #6-69  |
| X3CTL    | = 000004 R  | #7-80  |
| X3DEV    | = 000002 R  | #7-79  |
| X3PDF    | = 000000 R  | 9-113  |
| X3PKW    | = 000000 RG | 9-115      #10-133   |
| X3PST    | = 000000 RG | #10-133  |
| X3TMCT   | = 000007    | #6-70      11-225      11-241  |
| X3UNT    | = 000006 R  | #7-81  |
| X3.BEG   | = 000212 R  | #11-219  |
| X3.DBS   | = 000052 R  | #11-181  |
| X3.DWS   | = 000132 R  | #11-200  |
| X3.END   | = 000312 R  | #11-241  |
| X3.MBS   | = 000074 R  | #11-189  |
| X3.MWS   | = 000154 R  | #11-208  |
| X3.NXT   | = 000224 R  | #11-224  |
| \$ALPHA  | = 000022    | #10-133  |
| \$ANY    | = 000020    | #10-133  |
| \$BLANK  | = 000006    | #10-133  |
| \$DIGIT  | = 000024    | #10-133  |
| \$DNUMB  | = 000014    | #10-133  |
| \$EOS    | = 000012    | #10-133  |
| \$ERR1   | = ***** GX  | 9-127  |
| \$ERR2B  | = 000012 R  | #8-91      11-185  |
| \$ERR2C  | = ***** GX  | 11-195   |
| \$ERR2D  | = 000046 R  | #8-92      11-196  |
| \$ERR2E  | = 000122 R  | #8-93      11-204  |
| \$ERR2F  | = ***** GX  | 11-214   |
| \$ERR2G  | = 000156 R  | #8-94      11-215  |
| \$ERR2H  | = 000232 R  | #8-95      11-237  |
| \$EXIT   | = 000000    | #10-133  |
| \$FAIL   | = 177777    | #10-133  |
| \$GPRM   | = *****     | 10-133   |
| \$LAMDA  | = 000000    | #10-133  |
| \$NUMBER | = 000002    | #10-133  |

```

292 000602 062700 000002      ADD    #D$RNN,R0      ; * GET ADDRESS OF REMOTE LISTHEAD ADDRESS
293 000606 011000      MOV    (R0),R0      ; * GET ADDRESS OF REMOTE LISTHEAD
294 000610 010003      MOV    R0,R3      ; * SAVE ADDRESS OF LISTHEAD
295
296 000612 010301      14$: MOV    R3,R1      ; * SAVE UNMAPPED ADDRESS OF PREVIOUS BLOCK
297 000614 011000      MOV    (R0),R0      ; * GET NEXT BLOCK IN LIST
298 000616 001006      BNE    17$      ; * IF NOT END OF LIST - BRANCH
299 000620 010267      MOV    R2,$LSTRM      ; * UPDATE LAST NODE POINTER
300 000624 016767      MOV    RADDR,$LSTAD      ; * UPDATE HIGHEST NODE ADDRESS
301 000632 000411      BR     18$      ; * CONTINUE
302
303 000634 010C03      17$: MOV    R0,R3      ; * SAVE UNMAPPED ADDRESS
304 000636 010046      MOV    R0,-(SP)      ; * SET BUFFER ADDRESS
305 000640      CALL    $CEACX      ; * GET ACCESS TO BUFFER
306 000644 012600      MOV    (SP)+,R0      ; * GET MAPPED BUFFER ADDRESS
307 000646 026760      15$: CMP    RADDR,R.ADD(R0) ; * LINK REMOTE BLOCK HERE ?
308 000654 101356      BHI    14$      ; * BR IF NO
309 000656 010246      18$: MOV    R2,-(SP)      ; * GET ADDRESS OF BLOCK
310 000660 010146      MOV    R1,-(SP)      ; * GET ADDRESS OF PREVIOUS BLOCK
311 000662      CALL    $LINKX      ; * LINK BLOCK INTO LIST
312 000666      RESMAP      ; * RESTORE PREVIOUS MAPPING
313 000672      RETURN      ; * RETURN TO USER MODE AND TO CALLER
314
315 000674 103404      30$: BCS    111$      ; BR IF UNABLE TO ALLOCATE BLOCK
316 000676      RETURN
317
318      ; ERROR CONDITIONS
319
320 000700      101$: MSG$R YV      ; ILLEGAL NODE ADDRESS
321 000706      111$: MSG$R YV      ; REMOTE BLOCK ALLOCATION FAILURE
322      ; LET NTLUNL DE-ALLOCATE
323

```

```

111      ;
112      ; OBJ$DF ACTION ROUTINES
113      ;
114      ;
115      ; ** OBJECT NUMBER
116      ;
117 000000 116767 000000G 000000' 0.NUM:: MOVB .PNUMB,TYPE ; STORE OBJECT NUMBER
118 000006 005067 000002' CLR ONAME ; CLEAR THE OBJECT NAME
119 000012 005067 000004' CLR ONAME+2 ;
120 000016 105067 000006' CLR MBXCP ; CLEAR THE MAXIMUM NUMBER OF COPIES
121 000022 005767 000000G TST .PNUMH ; IS OBJECT NUMBER A WORD VALUE ?
122 000026 001004 BNE 101$ ; IF NE, NO
123 000030 105767 000001G TSTB .PNUMB+1 ; IS IT A BYTE VALUE ?
124 000034 001001 BNE 101$ ; IF NE, NO
125 000036 RETURN
126      ;
127      ; ERROR CONDITIONS
128      ;
129 000040 101$: MSG$R XW ; ILLEGAL OBJECT NUMBER
130      ;
131      ; ** OBJECT NAME
132      ;
133 000046 016700 000000G 0.NAME::MOV .PSTPT,RO ; GET ADDRESS OF ASCII TASK NAME
134 000052 010701 MOV PC,R1 ; PERIODS ARE ACCEPTABLE
135 000054 CALL $CAT5 ; CONVERT TASK NAME TO RAD50
136 000060 010167 000002' MOV R1,ONAME ; STORE THE OBJECT NAME
137 000064 103405 BCS 10$ ; IF CS, LESS THAN 3 CHARS CONVERTED
138 000066 010701 MOV PC,R1 ; CONVERT 2ND HALF OF NAME
139 000070 CALL $CAT5 ;
140 000074 010167 000004' MOV R1,ONAME+2 ; STORE 2ND HALF OF TASK NAME
141 000100 022767 000006 000000G 10$: CMP #6,.PSTCN ; IS THE TASK NAME TOO BIG ?
142 000106 103401 BLO 101$ ; IF LO, YES
143 000110 RETURN
144      ;
145      ; ERROR CONDITIONS
146      ;
147 000112 101$: MSG$R XX ; ILLEGAL TASK NAME
148      ;
149      ; ** VERIFICATION LEVEL
150      ;
151 000120 116767 000000G 000001' 0.VFY:: MOVB .PNUMB,FLAG ; SET VERIFICATION LEVEL
152 000126 022767 000010 000000G CMP #10,.PNUMB ; IS VERIFICATION LEVEL IN RANGE ?
153 000134 101401 BLOS 101$ ; IF LOS, NO
154 000136 RETURN
155      ;
156      ; ERROR CONDITIONS
157      ;
158 000140 101$: MSG$R XY ; ILLEGAL VERIFICATION LEVEL
159      ;
160      ; ** OBJECT FLAGS
161      ;
162 000146 156767 000000G 000001' 0.FLAG::BISB .PNUMB,FLAG ; SETUP OBJECT FLAGS
163 000154 032767 000007 000000G BIT #7,.PNUMB ; OVER-WRITING VERIFICATION LEVEL ?
164 000162 001004 BNE 101$ ; IF NE, YES
165 000164 105767 000001G TSTB .PNUMB+1 ; IS IT A BYTE VALUE ?
166 000170 001001 BNE 101$ ; IF NE, NO
167 000172 RETURN

```



```

193                                     .SBTTL  TPARS ACTION ROUTINES FOR ROU$DF
194
195
196 000224 012705 000000'      GROU::  MOV    #ROUDF,R5      ; SET UP STATE TABLE ADDRESS
197 000230 005001              CLR    R1                  ; FULL KEYWORD MATCH LENGTH
198 000232 012702 000000'      MOV    #ROUKW,R2          ; KEYWORD TABLE ADDRESS
199 000236 016703 000000G      MOV    CFGSZ,R3           ; RECORD LENGTH
200 000242 012704 000000G      MOV    #CFGBF,R4           ; RECORD BUFFER ADDRESS
201 000246 005067 000000G      CLR    SYNERR              ; CLEAR SYNTAX ERROR FLAG
202 000252              CALL    .TPARS                     ; PARSE LINE
203 000256 103003              BCC    20$                  ; IF SUCCESS - BRANCH
204 000260 005367 000000G      DEC    SYNERR              ; DID SYNTAX ERROR OCCUR ?
205 000264 001401              BEQ    101$                 ; IF YES - ERROR
206 000266              20$:  RETURN
207
208 000270              101$:  MSG$R  1T                    ; SYNTAX ERROR
209
210

```

77  
78  
79  
80  
81  
82  
83  
84  
85 000000  
86  
87  
88  
89  
90 000000  
91 000002  
92 000004  
93  
94 000014

```
.SBTTL  LOCAL DATA

;+
; LOCAL DATA
;-

.PSECT  DATA,D

; EVENT FILTER CONTROL BLOCK VALUES
CLASS:  .BLKW  1      ; EVENT CLASS AND FLAGS
FLAG:   .BLKW  1      ; FILTER BLOCK FLAGS WORD
EVENT:  .BLKW  4      ; EVENT TYPE MASK BITS

STRADD: .BLKW  1      ; ADDRESS OF STORAGE FOR EVENT TYPE MASK
```

```

87
88
89      ; SRHFNB - SEARCH FOR FILE NAME BLOCK
90
91      INPUTS --
92          FNBUFF CONTAINS FILE DESCRIPTOR INFORMATION
93
94      OUTPUTS --
95          C CLEAR - FILE NAME BLOCK FOUND
96                  - RO CONTAINS UNMAPPED BLOCK ADDRESS
97
98          C SET  - FILE NAME BLOCK NOT FOUND
99      -
100
101      SRHFNB: SWSTK$ 50$      ; ENTER SYSTEM STATE
102              MOV     @DECPT,R0      ; GET HOME BLOCK ADDRESS
103              ADD     #D$FNBR,R0     ; POINT AT FIRST FILE NAME BLOCK
104              MOV     @KSAR5,-(SP)    ; SAVE MAPPING
105
106              MOV     R0,R3          ; SAVE LISTHEAD
107      5$:      MOV     (R3),R0        ; GET ADDRESS OF FIRST BLOCK
108              BNE     10$           ; IF NOT END OF LIST BRANCH
109              RETC    R0            ; ELSE - ERROR
110              BR      40$           ; RETURN
111
112      10$:     MOV     R0,R2          ; SAVE UNMAPPED BLOCK ADDRESS
113              MOV     R0,-(SP)        ; STORE ADDRESS OF BLOCK
114              CALL    $CEACX         ; GET ACCESS TO BUFFER
115              MOV     (SP)+,R0        ; GET MAPPED ADDRESS TO BUFFER
116              MOV     R0,R3          ; SAVE MAPPED ADDRESS
117              MOV     #FNBUFF,R1     ; COPY ADDRESS OF LOCAL FILE BLOCK
118              ADD     #F.USE+1,R0     ; POINT TO FILESPEC DATA OF CURRENT BLOCK
119              ADD     #F.USE+1,R1     ; POINT TO FILESPACE DATA OF LOCAL BLOCK
120
121              MOV     #FIELDCL,R5    ; SET UP FIELD COUNT
122              CALL    CMPSTR          ; CHECK FOR FIELD EQUALITY
123              BCS     5$             ; IF NO MATCH - BRANCH
124
125              MOV     R2,4(SP)        ; IF MATCH - COPY UNMAPPED ADDRESS
126
127      40$:     MOV     (SP)+,@KSAR5   ; RESTORE MAPPING
128      50$:     RETURN
129
101 000000
102 000004 017700 000000G
103 000010 062700 000034
104 000014 017746 000000G
105
106 000020 010003
107 000022 011300
108 000024 001006
109 000026
110 000040 000423
111
112 000042 010002
113 000044 010046
114 000046
115 000052 012600
116 000054 010003
117 000056 012701 000000G
118 000062 062700 000003
119 000066 062701 000003
120
121 000072 012705 000000G
122 000076
123 000102 103747
124
125 000104 010266 000004
126
127 000110 012677 000000G
128 000114
129
  
```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL    | VALUE    | REFERENCES              |
|-----------|----------|-------------------------|
| \$\$\$BAS | = *****  | 6-80 6-80               |
| S.ADD     | 000004   | #6-68                   |
| S.CIR     | 000016   | #6-68                   |
| S.COST    | 000001   | #6-70                   |
| S.DEV     | 000020   | #6-68                   |
| S.DIA     | 000046   | #6-68 12-349            |
| S.DPA     | 000032   | #6-68                   |
| S.DPC     | 000036   | #6-68                   |
| S.DUM     | 000050   | #6-68 12-353            |
| S.FLAG    | 000002   | #6-68 *12-376           |
| S.FLG     | 000000   | #6-70                   |
| S.HAD     | 000010   | #6-68                   |
| S.HST     | 000006   | #6-68                   |
| S.LEN     | 000004   | #6-70                   |
| S.LENG    | 000052   | #6-68                   |
| S.LNK     | 000000   | #6-68                   |
| S.LOA     | 000044   | #6-68 12-345            |
| S.NMST    | 000002   | #6-70                   |
| S.OWNR    | 000003   | #6-70                   |
| S.PSS     | 000022   | #6-68                   |
| S.SEC     | 000040   | #6-68 12-337            |
| S.TER     | 000042   | #6-68 12-341            |
| UICBUF    | = *****  | GX 10-244 10-245 12-393 |
| UISZ      | = *****  | GX 12-394               |
| \$\$\$MCB | = *****  | 6-69 6-69               |
| ZF.COQ    | = 001000 | #6-69                   |
| ZF.DDM    | = 000001 | #6-69                   |
| ZF.DIA    | = 004000 | #6-69                   |
| ZF.DLC    | = 000002 | #6-69                   |
| ZF.DVP    | = 100000 | #6-69                   |
| ZF.INI    | = 040000 | #6-69                   |
| ZF.KMX    | = 000020 | #6-69                   |
| ZF.LLC    | = 000004 | #6-69                   |
| ZF.LMC    | = 000100 | #6-69                   |
| ZF.MAN    | = 020000 | #6-69                   |
| ZF.MFL    | = 000010 | #6-69                   |
| ZF.MTM    | = 000400 | #6-69                   |
| ZF.MUX    | = 000040 | #6-69                   |
| ZF.PSE    | = 002000 | #6-69                   |
| ZF.SLI    | = 010000 | #6-69                   |
| ZF.TIM    | = 000200 | #6-69                   |
| ZF.X3P    | = 000000 | #6-69                   |
| ZS.ASN    | = 100000 | #6-69                   |
| ZS.BSY    | = 140000 | #6-69                   |
| Z.AVL     | 000014   | #6-69                   |
| Z.DAT     | 000016   | #6-69                   |
| Z.DSP     | 000000   | #6-69 6-69              |
| Z.FLG     | 000010   | #6-69                   |
| Z.LEN     | = 000016 | #6-69                   |
| Z.LLN     | 000006   | #6-69                   |
| Z.MAP     | 000020   | #6-69                   |
| Z.NAM     | 000004   | #6-69                   |

```

302      ; LINE NAME (SLT$DF)
303
304      S.NAME::MOV    .PSTPT,RO      ; GET TPARS POINTER
305      000660 016700 000000G      MOV    PC,R1      ; ALLOW ... AS CHARACTER
306      000664 010701              CALL    $CAT5      ; CONVERT TO RAD50
307      000666              MOV    R1,NAME      ; SAVE LINE (DDM) NAME
308      000672 010167 000000'      RETURN
309      000676
310
311      ; CONTROLLER NUMBER (SLT$DF)
312
313      S.CTL::MOVB    .PNUMB,CTL      ; SAVE NUMBER
314      000700 116767 000000G 000003'
315      000706              RETURN
316
317      ; UNIT NUMBER (SLT$DF)
318
319      S.UNT::MOV    @SLTMA,R0      ; GET SLT MAPPING TABLE ADDRESS
320      000710 017700 000000G      MOV    @SLTNM,R1      ; GET NUMBER OF SLT'S
321      000714 017701 000000G      BEQ    40$      ; IF ZERO, LINE NOT FOUND !
322      000720 001426              10$: MOV    (R0)+,R2      ; GET NEXT SLT
323      000722 012002              BEQ    30$      ; IF EMPTY, SKIP IT
324      000724 001422
325
326      000726 126267 000012 000003' 15$: CMPB    L.CTL(R2),CTL      ; CONTROLLER NUMBER MATCH ?
327      000734 001016              BNE    30$      ; IF NE, NO
328      000736 126267 000013 000000G  CMPB    L.UNT(R2),.PNUMB      ; UNIT NUMBER MATCH ?
329      000744 001012              BNE    30$      ; IF NE, NO
330      000746 116202 000002          MOVB    L.DDM(R2),R2      ; GET DDM PDV INDEX
331      000752 067702 000000G      ADD    @PDVTA,R2      ; PLUS PDV TABLE ADDRESS
332      000756 016202 000000G      MOV    .CBIAS(R2),R2      ; GET PDV ADDRESS
333
334      000762 026267 000000' 000000' 20$: CMP    Z.NAM(R2),NAME      ; DO NAMES MATCH ?
335      000770 001405              BEQ    50$      ; IF EQ, YES
336      000772              30$: SDB    R1,10$      ; LOOP <SLT COUNT> TIMES
337      000776              40$: MSG$R  XP      ; ERROR - LINE NOT IN SYSTEM
338      001004 014002              50$: MOV    -(R0),R2      ; PICK UP SLT ADDRESS
339
340      001006 010267 000000G      MOV    R2,$SLTA      ; SAVE IT
341      001012 005067 000000G      CLR    STAN      ; INITIALIZE STATIONS FOR THIS LINE
342      001016 056762 000000G 000000  BIS    MRKFL,L.FLG(R2)      ; UPDATE MARK-FOR-LOAD STATUS
343      001024 032762 000400 000000  BIT    #LF.BRO,L.FLG(R2)      ; BROADCAST LINE ?
344      001032 001424              BEQ    55$      ; IF NOT - BRANCH
345      001034 032762 002000 000000  BIT    #LF.ENA,L.FLG(R2)      ; IS LINE ENABLED ?
346      001042 001406              BEQ    52$      ; IF NOT - BRANCH
347      001044 062702 000022          ADD    #L.MPF,R2      ; POINT TO FIRST TRIB FLAG
348      001050 152712 000100          BISB   #SF.ENA,(R2)      ; ENABLE ETHERNET CIRCUIT
349      001054 016702 000000G      MOV    $SLTA,R2      ; RESTORE SLT ADDRESS
350      001060 032762 000040 000000 52$: BIT    #LF.SER,L.FLG(R2)      ; SERVICE DISABLE ?
351      001066 001406              BEQ    55$      ; IF NOT - BRANCH
352      001070 062702 000022          ADD    #L.MPF,R2      ; POINT TO EXTENSION
353      001074 152712 000001          BISB   #SF.SER,(R2)      ; SET SERVICE DISABLE
354
355      001100 016702 000000G      MOV    $SLTA,R2      ; RESTORE SLT ADDRESS
356      001104 032762 004000 000000 55$: BIT    #LF.MFL,L.FLG(R2)      ; IS LINE MARKED-FOR-LOAD ?
357      001112 001003              BNE    60$      ; IF NE, YES ... OKAY
358      001114 042762 002000 000000  BIC    #LF.ENA,L.FLG(R2)      ; ELSE, MAKE SURE ENABLE ISN'T SET

```

```

131 000202 005302          DEC      R2          ; SET FLAG FOR SECOND ADJACENCY
132 000204 000742          BR       3$          ; CONTINUE
133
134 000206 016700 000000G    1$:  MOV     XTLLN,R0          ; STORE NUMBER OF XPT CIRCUITS
135 000212 012701 000046    MOV     #T$LEN,R1         ; STORE COUNTER BLOCK SIZE
136 000216                CALL    $MUL             ; CALCULATE TOTAL AMOUNT
137 000222 062701 000077    ADD     #77,R1           ; CALCULATE BLOCKS NEEDED
138 000226 042701 000077    BIC     #77,R1           ; ...
139 000232 006201          ASR     R1               ; ...
140 000234 006201          ASR     R1               ; ...
141 000236 006201          ASR     R1               ; ...
142 000240 006201          ASR     R1               ; ...
143 000242 006201          ASR     R1               ; ...
144 000244 006201          ASR     R1               ; ...
145
146 000246 060167 000000G    ADD     R1,.,NOMAP        ; ADD TO TOTAL OF NON UMR MAPPED POOL..
147 000252 060167 000000G    ADD     R1,.,FUDGE       ; ADD TO TOTAL POOL-(RDB'S+SDB'S)
148
149 000256 016700 000000G    MOV     ECLLN,R0          ; COPY NUMBER OF MAX LOGICAL LINKS
150 000262 012701 000124    MOV     #L.LNG,R1         ; GET SIZE OF LOGICAL LINK
151 000266                CALL    $MUL             ; CALCULATE TOTAL SIZE
152 000272 062701 000077    ADD     #77,R1           ; ROUND UP TO NEXT BLOCK
153 000276 006201          ASR     R1               ; CONVERT TO BLOCKS
154 000300 006201          ASR     R1               ; ...
155 000302 006201          ASR     R1               ; ...
156 000304 006201          ASR     R1               ; ...
157 000306 006201          ASR     R1               ; ...
158 000310 006201          ASR     R1               ; ...
159 000312 060167 000000G    ADD     R1,.,ALLOC       ; ADJUST BYTE POOL SIZE
160 000316 060167 000000G    ADD     R1,.,FUDGE       ; ADJUST PARTITION SIZE - (RDB+SDB'S)
161
162 000322 016700 000000G    MOV     ECLNOD,R0         ; COPY ECL NODE COUNTERS VALUE
163 000326 012701 000050    MOV     #E$NLEN,R1        ; GET NODE COUNTER SIZE
164 000332                CALL    $MUL             ; CALCULATE TOTAL BYTES
165 000336 062701 000077    ADD     #77,R1           ; ROUND UP TO NEXT BLOCK
166 000342 042701 000077    BIC     #77,R1           ; ...
167 000346 006201          ASR     R1               ; CONVERT TO BLOCKS
168 000350 006201          ASR     R1               ; ...
169 000352 006201          ASR     R1               ; ...
170 000354 006201          ASR     R1               ; ...
171 000356 006201          ASR     R1               ; ...
172 000360 006201          ASR     R1               ; ...
173 000362 060167 000000G    ADD     R1,.,NOMAP        ; ADJUST NON UMR BLOCK POOL
174 000366 060167 000000G    ADD     R1,.,FUDGE       ; ADJUST POOL SIZE
175
176                .IF DF R$S11M
177 000372 032777 000000G 000000G    BIT     #FE.EXT,@FMASK ; IS THIS REALLY AN 11/70 ?
178 000400 001002          BNE     2$                ; IF NE, YES
179 000402 005267 000000G    INC     ,UMRFL          ; DISABLE UMR ALLOCATION
180 000406 016701 000000G    . 1$: MOV     ,PCB,R1         ; WAS FIRST PARTITION FOUND ?
181 000412 001103          BNE     90$              ; IF NE, YES
182 000414 005000          CLR     R0                ; INDICATE NO ERROR YET
183 000416 012702 000000G    MOV     #,PNAM1,R2        ; POINT AT FIRST NAME
184 000422 005767 000000G    TST     ,PNAM2          ; WAS SECOND NAME SPECIFIED ?
185 000426 001003          BNE     5$                ; IF NE, YES
186 000430          MSG$R    XC                      ; ERROR - PARTITION NOT IN SYSTEM
187 000436 005712    5$:  TST     (R2)              ; WAS FIRST NAME SPECIFIED ?

```

CF2AC8 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:12  
Table of contents

|     |     |                                  |
|-----|-----|----------------------------------|
| 5-  | 45  | MACRO DEFINITIONS                |
| 6-  | 91  | SERSLN - DETERMINE SLN/STA VALUE |
| 9-  | 252 | PAS - SERVICE BLOCK PASSWORD     |
| 11- | 372 | FSLT - FIND SLT ADDRESS          |

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL | VALUE       | REFERENCES                     |
|--------|-------------|--------------------------------|
| CERR   | = ***** GX  | 5-85                           |
| CLIN   | = ***** GX  | 5-85                           |
| CIRMUL | = ***** GX  | 6-119                          |
| CIRNAM | = ***** GX  | 6-106 10-328                   |
| CIRNUM | = ***** GX  | 6-113                          |
| CIRUNT | = ***** GX  | 6-115                          |
| DECP   | = ***** GX  | 8-228                          |
| D\$SER | = 000032    | 8-229                          |
| ENDHEX | = ***** GX  | 9-269                          |
| ENDPAS | = ***** GX  | 9-296                          |
| FMT8   | = ***** GX  | 5-85                           |
| FM.8   | = 000000    | #5-85                          |
| FDPV   | = 000656 R  | 6-108 #10-350                  |
| FSLT   | = 000724 R  | 6-116 #11-387                  |
| F.NXT  | = 000000    | #5-59                          |
| F.USE  | = 000002    | #5-59                          |
| F.VAR  | = 000003    | #5-59                          |
| HEXIND | = 000604 R  | 8-159 8-167 9-285 9-287 #9-307 |
| HEXSTR | = ***** GX  | 9-267 9-272 9-283              |
| HRDADD | = 000074 RG | #8-156                         |
| I\$BAS | = *****     | 5-61                           |
| KSARS  | = ***** GX  | 8-194 8-245 9-309              |
| LET    | = 000100    | #5-78                          |
| LF.ACT | = 100000    | #5-62                          |
| LF.BRO | = 000400    | #5-62                          |
| LF.BWT | = 000007    | #5-62                          |
| LF.ENA | = 002000    | #5-62                          |
| LF.LPB | = 001000    | #5-62                          |
| LF.MDC | = 000100    | #5-62                          |
| LF.MFL | = 004000    | #5-62                          |
| LF.MTP | = 000020    | #5-62                          |
| LF.PAC | = 000200    | #5-62                          |
| LF.RDY | = 040000    | #5-62                          |
| LF.REA | = 010000    | #5-62                          |
| LF.SER | = 000040    | #5-62                          |
| LF.TIM | = 000010    | #5-62                          |
| LF.UNL | = 020000    | #5-62                          |
| LF.X2P | = 000000    | #5-62                          |
| LN.CLO | = 000000    | #5-62                          |
| LN.DUM | = 000005    | #5-62                          |
| LN.LOA | = 000004    | #5-62                          |
| LN.LOO | = 000003    | #5-62                          |
| LN.OAU | = 000003    | #5-62                          |
| LN.OFF | = 000001    | #5-62                          |
| LN.ON  | = 000000    | #5-62                          |
| LN.OOP | = 000004    | #5-62                          |
| LN.OPE | = 000001    | #5-62                          |
| LN.REF | = 000002    | #5-62                          |
| LN.SER | = 000002    | #5-62                          |
| LN.STA | = 000017    | #5-62                          |
| LN.SUB | = 000360    | #5-62                          |
| LN.TRI | = 000006    | #5-62                          |



CF2AC9 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:13 Page 7-3

```

ACTION ROUTINES FOR PAR$DF

248 000462 103403          BCS      30$          ; IF CS, N-U-M AREA > PARTITION !!
249 000464 026700 000000G  CMP      NTPSZ,R0      ; ENOUGH LEFT FOR RDB/SDB ?
250 000471 101403          BLOS     40$          ; IF LOS, YES
251
252 000472          30$:    MSG$R   XF          ; NON UMR POOL LARGER THAN PARTITION
253
254 000500          40$:    RETURN
255          .ENDC
256
257          .IF DF M$$MGE
258 000502 005267 000000G  P.NONE: INC      .UMRFL      ; DISABLE UMR ALLOCATION
259 000506          RETURN
260          .ENDC
261

```

## SYMBOL CROSS REFERENCE

CREF    04.00

SYMBOL    VALUE      REFERENCES

|         |          |              |        |        |       |       |                  |
|---------|----------|--------------|--------|--------|-------|-------|------------------|
| SF.ACT  | = 000200 | #6-66        |        |        |       |       |                  |
| SF.CIR  | = 000010 | #6-64        |        |        |       |       |                  |
| SF.DEV  | = 000020 | #6-64        |        |        |       |       |                  |
| SF.DIA  | = 004000 | #6-64        |        |        |       |       |                  |
| SF.DPA  | = 000100 | #6-64        |        |        |       |       |                  |
| SF.DPC  | = 000200 | #6-64        |        |        |       |       |                  |
| SF.DUM  | = 010000 | #6-64        |        |        |       |       |                  |
| SF.ENA  | = 000100 | #6-66        |        |        |       |       |                  |
| SF.HAD  | = 000004 | #6-64        |        |        |       |       |                  |
| SF.HST  | = 000002 | #6-64        |        |        |       |       |                  |
| SF.LOA  | = 002000 | #6-64        |        |        |       |       |                  |
| SF.LPB  | = 000004 | #6-66        |        |        |       |       |                  |
| SF.MFL  | = 000040 | #6-66        |        |        |       |       |                  |
| SF.PAC  | = 000020 | #6-66        |        |        |       |       |                  |
| SF.PHS  | = 020000 | #6-64        |        |        |       |       |                  |
| SF.PSS  | = 000040 | #6-64        |        |        |       |       |                  |
| SF.REA  | = 000010 | #6-66        |        |        |       |       |                  |
| SF.SEC  | = 000400 | #6-64        |        |        |       |       |                  |
| SF.SER  | = 000001 | #6-66        |        |        |       |       |                  |
| SF.SVC  | = 000002 | #6-66        |        |        |       |       |                  |
| SF.TER  | = 001000 | #6-64        |        |        |       |       |                  |
| SF.UNL  | = 000040 | #6-66        |        |        |       |       |                  |
| STRADD  | = *****  | GX    8-276  | 8-299  |        |       |       |                  |
| STRPT   | = *****  | GX    *8-281 | *8-312 |        |       |       |                  |
| STRSIZ  | = *****  | GX    8-275  | *8-279 | *8-282 | 8-297 | 8-307 | *8-309    *8-313 |
| SYNERR  | = *****  | GX    *7-96  | *7-99  |        |       |       |                  |
| S.ADD   | 000004   | #6-64        |        |        |       |       |                  |
| S.CIR   | 000016   | #6-64        |        |        |       |       |                  |
| S.COST  | 000001   | #6-66        |        |        |       |       |                  |
| S.DEV   | 000020   | #6-64        |        |        |       |       |                  |
| S.DIA   | 000046   | #6-64        |        |        |       |       |                  |
| S.DPA   | 000032   | #6-64        |        |        |       |       |                  |
| S.DPC   | 000036   | #6-64        |        |        |       |       |                  |
| S.DUM   | 000050   | #6-64        |        |        |       |       |                  |
| S.FLAG  | 000002   | #6-64        |        |        |       |       |                  |
| S.FLG   | 000000   | #6-66        |        |        |       |       |                  |
| S.HAD   | 000010   | #6-64        |        |        |       |       |                  |
| S.HST   | 000006   | #6-64        |        |        |       |       |                  |
| S.LEN   | 000004   | #6-66        |        |        |       |       |                  |
| S.LENG  | 000052   | #6-64        |        |        |       |       |                  |
| S.LNK   | 000000   | #6-64        |        |        |       |       |                  |
| S.LOA   | 000044   | #6-64        |        |        |       |       |                  |
| S.NMST  | 000002   | #6-66        |        |        |       |       |                  |
| S.OWNR  | 000003   | #6-66        |        |        |       |       |                  |
| S.PSS   | 000022   | #6-64        |        |        |       |       |                  |
| S.SEC   | 000040   | #6-64        |        |        |       |       |                  |
| S.TER   | 000042   | #6-64        |        |        |       |       |                  |
| XPCIR   | 001026   | RG    *9-380 |        |        |       |       |                  |
| XPTDF   | 000142   | R      7-87  |        |        |       |       |                  |
| XPTLLN  | = *****  | GX    *9-380 |        |        |       |       |                  |
| X\$EMCB | = *****  |              | 6-65   |        |       |       |                  |
| ZF.COUI | = 001000 | #6-65        |        |        |       |       |                  |

|                  |                     |                |                  |                   |
|------------------|---------------------|----------------|------------------|-------------------|
| ASSCHK= 000000   | D\$IPL 000051       | E\$XPR= 000000 | N\$BUI= 000001   | \$EOS = 000012    |
| ASSCPS= 000000   | D\$IID 000020       | F\$LVL= 000001 | N\$LDV= 000001   | \$ERRXO= ***** GX |
| ASSPRI= 000000   | D\$INAM 000006      | G\$TTP= 000000 | N\$MCP= 000001   | \$ERRXP= ***** GX |
| ASSTRP= 000000   | D\$INUM 000014      | G\$TSS= 000000 | N\$MML= 000001   | \$ERRXQ= ***** GX |
| CFGBF = ***** GX | D\$IST 000047       | G\$TTK= 000000 | N\$MOV= 000010   | \$ERRXR= ***** GX |
| CFGSZ = ***** GX | D\$MAXC 000064      | G\$WRD= 000000 | N\$NCT= 000001   | \$ERRXS= ***** GX |
| C\$CKP= 000000   | D\$MAXH 000066      | INAC 000226RG  | N\$PEM= 000001   | \$ERRXT= ***** GX |
| C\$DRE= 000400   | D\$MAXV 000070      | INCT 000146RG  | OUTT 000176RG    | \$ERRXU= ***** GX |
| C\$RSH= 177564   | D\$MLL 000040       | IPL 000336RG   | P\$P45= 000000   | \$ERRXW= ***** GX |
| DECD 000000R     | 002 D\$MNOD 000041  | I\$RAR= 000000 | P\$WRD= 000000   | \$ERRXX= ***** GX |
| DECD1 000014R    | 002 D\$NA 000062    | I\$RDN= 000000 | Q\$OPT= 000010   | \$ERRXY= ***** GX |
| DECKW 000000RG   | 003 D\$NBEA 000056  | K\$CNT= 177546 | REF 000416RG     | \$ERR1T= ***** GX |
| DECP = ***** GX  | D\$NBRA 000054      | K\$CSR= 177546 | R\$DER= 000000   | \$EXIT = 000000   |
| DECST 000000RG   | 002 D\$NEND= 000054 | K\$LDL= 000000 | R\$K11= 000001   | \$FAIL = 177777   |
| DEL 000256RG     | D\$NLN 000030       | K\$TPS= 000074 | R\$SND= 000000   | \$LAMD= 000000    |
| DELW 000306RG    | D\$NN 000060        | LDLP = 000000  | R\$11M= 000000   | \$NUMBR= 000002   |
| D\$AMXC 000072   | D\$DUT 000043       | LST 000366RG   | SEG 000446RG     | \$QDEC 000000RG   |
| D\$AMXH 000074   | D\$RETF 000050      | L\$ASG= 000000 | SYNERR= ***** GX | \$RAD50= 000016   |
| D\$ANN 000000    | D\$RNN 000002       | L\$DRV= 000000 | \$SWRG= 000000   | \$STRNG= 000004   |
| D\$BRPR 000102   | D\$RTMR 000076      | L\$P11= 000001 | \$SYSZ= 007600   | \$SUBXP= 000010   |
| D\$BTM 000100    | D\$SEG 000036       | L\$11R= 000000 | T\$KMG= 000000   | \$LINX= ***** GX  |
| D\$DEL 000045    | D\$SER 000032       | M\$CRB= 000124 | T\$MIN= 000000   | \$NODC= ***** GX  |
| D\$DELW 000046   | D\$SRL 000052       | M\$CRX= 000000 | V\$CTR= 001000   | \$FLG= 177777     |
| D\$END = 000104  | D\$BUG= 177514      | M\$FCS= 000000 | X\$DBT= 000000   | \$KEY= 000000     |
| D\$FNB 000034    | D\$ISK= 000000      | M\$MGE= 000000 | \$ALPHA= 000022  | \$STA= 000000     |
| D\$HIDR 000024   | D\$L11= 000001      | M\$NET= 000000 | \$ANY = 000020   | \$TMP= 000000R    |
| D\$HOST 000022   | D\$YNC= 000000      | M\$DVR= 000000 | \$BLANK= 000006  | .PNUMB= ***** GX  |
| D\$INAC 000044   | D\$YNM= 000000      | NODCT 000110RG | \$DIGIT= 000024  | .TPARS= ***** GX  |
| D\$INCT 000042   | ECLLNK 000052RG     | N\$ACC= 000001 | \$DNUMB= 000014  |                   |

. ABS. 000104 000 (RW,I,GBL,ABS,DVR)  
 000462 001 (RW,I,LCL,REL,CON)  
 \$STATE 000116 002 (RW,D,LCL,REL,CON)  
 \$TAB 000002 003 (RW,D,LCL,REL,CON)  
 \$KSTR 000007 004 (RW,D,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

work file reads: 0  
 work file writes: 0  
 Size of work file: 11923 Words ( 47 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:34.01

SY:CF2DEC.V2,[132,134]CF2DEC/CR/-SP=SY:[1,1]RSXMCML/ML,[130,110]NETLIB/ML,[130,10]RSXMCML/PA:1,[132,10]CF2DEC

CF2UMR CREATED BY MACRO ON 29-JUN-85 AT 00:14 PAGE 1 L 16  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE       | REFERENCES                        |
|---------|-------------|-----------------------------------|
| ALOCB   | = ***** GX  | 5-92                              |
| BUFUMP  | = 172354    | #5-64                             |
| CMODE   | = 140000    | #5-64                             |
| DEACB   | = ***** GX  | 5-167                             |
| FE_EXT  | = ***** GX  | 5-86                              |
| FMASK   | = ***** GX  | 5-86                              |
| IS\$AS  | = *****     | 5-60                              |
| KISAR0  | = 172340    | #5-64                             |
| KISAR6  | = 172354    | #5-64                             |
| MPAR    | = 172100    | #5-64                             |
| MPCSR   | = 177746    | #5-64                             |
| M\$MGE  | = 000000    | 5-64                              |
| M.BFVH  | = ***** GX  | 5-84 6-185                        |
| M.BFVL  | = ***** GX  | 5-111 6-184                       |
| M.LGTH  | = ***** GX  | 5-91 5-165                        |
| M.UMRA  | = ***** GX  | 5-147 6-186                       |
| M.UMRN  | = ***** GX  | 5-105 6-188                       |
| PIRQ    | = 177772    | #5-64                             |
| PMODE   | = 030000    | #5-64                             |
| PR0     | = 000000    | #5-64                             |
| PR1     | = 000040    | #5-64                             |
| PR2     | = 000100    | #5-64                             |
| PR3     | = 000140    | #5-64                             |
| PR4     | = 000200    | #5-64                             |
| PR5     | = 000240    | #5-64                             |
| PR6     | = 000300    | #5-64                             |
| PR7     | = 000340    | #5-64                             |
| PS      | = 177776    | #5-64                             |
| P.BLKS  | = ***** GX  | 5-99                              |
| P.REL   | = ***** GX  | 5-106                             |
| R\$MPL  | = *****     | 5-64 5-64 5-115 5-138 5-157 7-229 |
| R\$11D  | = *****     | 5-64                              |
| R\$11M  | = 000000    | 5-84 6-182                        |
| SWR     | = 177570    | #5-64                             |
| TPS     | = 177564    | #5-64                             |
| UBMPR   | = 170200    | #5-64 5-148                       |
| UISAR0  | = 177640    | #5-64                             |
| UISAR1  | = 177642    | #5-64                             |
| UMRLD   | = 000244 R  | 5-151 #6-183                      |
| \$ASUMR | = ***** GX  | 5-128                             |
| \$ERRXL | = ***** GX  | 5-156                             |
| \$ERRXM | = ***** GX  | 5-168                             |
| \$SAVRG | = ***** GX  | 5-98                              |
| \$UMRAL | = 000000 RG | #5-85                             |
| .ALLOC  | = ***** GX  | 5-100                             |
| .BASEB  | = 140000    | #5-64                             |
| .NOMAP  | = ***** GX  | 5-101                             |
| .PUMR   | = ***** GX  | 5-150                             |
| .UMRBA  | = ***** GX  | 5-146                             |
| .UMRFL  | = ***** GX  | 5-88                              |

••FILE••ID••CFGX3P

```

CCCCCCCC  FFFFFFFF  GGGGGGGG  XX  XX  333333  PPPPPPPP
CCCCCCCC  FFFFFFFF  GGGGGGGG  XX  XX  333333  PPPPPPPP
CC         FF         GG         XX  XX  33  33  PP  PP
CC         FF         GG         XX  XX  33  33  PP  PP
CC         FF         GG         XX  XX  33  33  PP  PP
CC         FF         GG         XX  XX  33  33  PP  PP
CC         FFFFFFFF  GG         XX  XX  33  33  PPPPPPPP
CC         FFFFFFFF  GG         XX  XX  33  33  PPPPPPPP
CC         FF         GG  GGGGGG  XX  XX  33  33  PP
CC         FF         GG  GGGGGG  XX  XX  33  33  PP
CC         FF         GG  GG      XX  XX  33  33  PP
CC         FF         GG  GG      XX  XX  33  33  PP
CC         FF         GG  GG      XX  XX  33  33  PP
CCCCCCCC  FF         GGGGGG  XX  XX  333333  PP
CCCCCCCC  FF         GGGGGG  XX  XX  333333  PP

```

```

LL         SSSSSSSS  TTTTTTTTTT
LL         SSSSSSSS  TTTTTTTTTT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LL         SSSSSS    TT
LL         SSSSSS    TT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LL         SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT

```

CFGX3P

SYMBOL

SYMBOL

\$QX3P

\$RAD5

\$RONI

\$STRN

\$SUB:

\$X3C

\$X3D

\$X3D

\$X3M

\$X3M

\$X3M

\$X3M

.PNUT

.PNUT

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

.TPA

CFGX3P      CREATED BY MACRO ON 29-JUN-85 AT 00:09      PAGE 2      M 2

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE      | REFERENCES          |
|---------|------------|---------------------|
| \$QX3P  | 000000 RG  | #9-117              |
| \$RAD50 | = 000016   | #10-133             |
| \$RONLY | = *****    | 10-133      10-133  |
| \$STRNG | = 000004   | #10-133             |
| \$SUBXP | = 000010   | #10-133             |
| \$X3CAL | = ***** GX | 11-219              |
| \$X3DBS | = ***** GX | *11-185      11-191 |
| \$X3DWS | = ***** GX | *11-202      11-210 |
| \$X3MBS | = ***** GX | *11-195             |
| \$X3MWS | = ***** GX | *11-212             |
| \$SFLG  | = 177777   | #10-133             |
| \$SKEY  | = 177777   | #10-133             |
| .PNUMB  | = ***** GX | 11-229      11-231  |
| .PNUMH  | = ***** GX | 11-227              |
| .TPARS  | = ***** GX | 9-119               |

```

325 .SBTTL COUNTER TIMER ACTION ROUTINE
326
327
328 ; COUNTER TIMER VALUE
329 ;
330 S.CTIM::
331 000714 005767 000000G TST .PNUMH ; LEGAL COUNTER TIMER VALUE
332 000720 001004 BNE 101$ ; BR IF NO
333 000722 016767 000000G 000000G MOV .PNUMB,$CTIM ; STORE COUNTER TIMER VALUE
334 000730 RETURN
335 000732 101$: MSG$R YN ; ILLEGAL COUNTER TIMER VALUE
336
337 000740 022767 000041 000000G S.BRA:: CMP #33,..PNUMB ; LEGAL MAX LIMIT ?
338 000746 002404 BLT 101$ ; IF NOT - BRANCH
339 000750 066767 000000G 000000G ADD .PNUMB,NBRA ; ADJUST TOTAL OF BROADCAST ROUTER ADJ.
340 000756 RETURN
341
342 000760 062716 000002 101$: ADD #2,(SP) ; REJECT PARAMETER
343 000764 RETURN
344

```

```

168
169
170
171 000174
172
173
174
175
176 000202 116767 000000G 000006' 0.CPY:: MOV .ENABL LSB ; GET NUMBER OF COPIES
177
178 000210 005767 000000G TST .PNUMH ; IS IT A WORD VALUE ?
179 000214 001076 BNE 101$ ; IF NE, NO
180 000216 105767 000001G TSTB .PNUMB+1 ; IS IT A BYTE VALUE ?
181 000222 001073 BNE 101$ ; IF NE, NO
182
183 000224 012701 000012 0.END:: MOV #0.LEN,R1 ; GET LENGTH OF AN OBJECT BLOCK
184
185 000230 SWSTK$ 30$ ; * ENTER KERNEL MODE
186 000234 CALL $XALOC ; * ATTEMPT TO ALLOCATE BLOCK
187 000240 103006 BCC 10$ ; * BR IF SUCCESS
188 000242 RETC R0 ; * ELSE SET USER C-BIT
189 000254 000453 BR 20$ ; * AND EXIT
190 000256 10$: SAVMAP ; * SAVE CURRENT MAPPING
191 000262 010002 MOV R0,R2 ; * SAVE BUFFER ADDRESS
192 000264 010046 MOV R0,-(SP) ; * SET BUFFER ADDRESS
193 000266 CALL $CEACX ; * GET ACCESS TO THE BLOCK
194 000272 012600 MOV (SP)+,R0 ; * GET MAPPED ADDRESS FOR BLOCK
195 000274 116760 000000' 000002 MOVB TYPE,0.TYP(R0) ; * STORE THE OBJECT TYPE NUMBER
196 000302 016760 000002' 000006 MOV ONAME,0.NAM(R0) ; * STORE THE OBJECT NAME
197 000310 016760 000004' 000010 MOV ONAME+2,0.NAM+2(R0) ; *
198 000316 116760 000001' 000003 MOVB FLAG,0.FLG(R0) ; * STORE THE FLAGS BYTE
199 000324 116760 000006' 000004 MOVB MXCP,0.MXC(R0) ; * COPY THE MAXIMUM NUMBER OF COPIES TO SPAWN
200 000332 016700 000000G MOV OBJHD,R0 ; * GET ADDRESS OF BEGINNING OF LIST
201 000336 010003 MOV R0,R3 ; * SAVE ADDRESS OF PREVIOUS BLOCK
202 000340 010301 14$: MOV R3,R1 ; * SAVE UNMAPPED ADDRESS OF PREVIOUS BLOCK
203 000342 011000 MOV (R0),R0 ; * GET NEXT BLOCK IN LIST
204 000344 001411 BEQ 18$ ; * IF EQ, AT END OF LIST
205 000346 010003 MOV R0,R3 ; * SAVE UNMAPPED ADDRESS
206 000350 010046 MOV R0,-(SP) ; * SET BLOCK ADDRESS
207 000352 CALL $CEACX ; * GET ACCESS TO BLOCK
208 000356 012600 MOV (SP)+,R0 ; * GET MAPPED ADDRESS OF BLOCK
209 000360 126760 000000' 000002 15$: CMPB TYPE,0.TYP(R0) ; * INSERT OBJECT TABLE ENTRY HERE ?
210 000366 101364 BHI 14$ ; * BR IF NO
211 000370 101246 18$: MOV R2,-(SP) ; * GET ADDRESS OF BLOCK TO LINK
212 000372 010146 MOV R1,-(SP) ; * GET ADDRESS OF PREVIOUS BLOCK IN LIST
213 000374 CALL $LINKX ; * LINK BLOCK INTO LIST
214 000400 RESMAP ; * RESTORE PREVIOUS MAPPING
215 000404 20$: RETURN ; * RETURN TO USER MODE AND TO CALLER
216
217 000406 103404 30$: BCS 111$ ; BR IF UNABLE TO ALLOCATE BLOCK
218 000410 RETURN
219
220
221
222
223 000412 101$: MSG$R YZ ; ILLEGAL NUMBER OF COPIES
224 000420 111$: MSG$R YY ; OBJECT BLOCK ALLOCATION FAILURE
; LET NTLUNL DE-ALLOCATE

```



```

212
213
214
215
216
217
218 000276 016767 000000G 000000G R.ADDR::MOV .PNUMB,,NN ; STORE THE MAXIMUM NUMBER OF NODES
219 ; ANY 16-BIT NUMBER LEGAL
220 000304 RETURN
221
222 000306 016767 000000G 000000G R.COST::MOV .PNUMB,,MAXC ; STORE THE MAXIMUM COST
223 000314 022767 001776 000000G CMP #1022,,.PNUMB ; VALUE IN RANGE ?
224 000322 103401 BLO 101$ ; IF LO, NO
225 000324 RETURN
226
227 000326 101$: MSG$R YK ; ILLEGAL COST
228
229 000334 016767 000000G 000000G R.HOP::MOV .PNUMB,,MAXH ; GET THE MAXIMUM NUMBER OF HOPS ALLOWED
230 000342 022767 000036 000000G CMP #30,,.MAXH ; VALUE IN RANGE ?
231 000350 103401 BLO 101$ ; IF LO, NO
232 000352 RETURN
233
234 000354 101$: MSG$R YL ; ILLEGAL HOP
235
236
237 000362 005767 000000G R.RTM::TST .PNUMH ; LEGAL ROUTING TIMER INTERVAL VALUE?
238 000366 001004 BNE 101$ ; BR IF NO
239 000370 016767 000000G 000000G MOV .PNUMB,,ROUTM ; GET ROUTING TIMER INTERVAL VALUE
240 000376 RETURN
241
242 000400 101$: MSG$R YW ; ILLEGAL ROUTING TIMER INTERVAL VALUE
243
244
245 000406 005767 000000G R.MAXA::TST .NN ; ROUTING NODE ?
246 000412 001004 BNE 10$ ; IF YES - BRANCH
247 000414 005767 000000G TST .PNUMB ; IF NO - VALUE MUST BE ZERO
248 000420 001405 BEQ 20$ ; IF LEGAL - BRANCH
249 000422 000410 BR 101$ ; ELSE - REPORT ERROR
250
251 000424 022767 000077 000000G 10$: CMP #63,,.PNUMB ; CHECK FOR HIGH LIMIT
252 000432 103404 BLO 101$ ; I. OUT OF RANGE -- BRANCH
253
254 000434 016767 000000G 000000G 20$: MOV .PNUMB,,NA ; GET NUMBER OF AREAS IN NETWORK
255 000442 RETURN
256
257 000444 101$: MSG$R YM ; ILLEGAL MAX AREA VALUE

```

```
96                                     .SBTTL  ERROR MESSAGES
97
98                                     ;
99                                     ; ERROR MESSAGES
100                                    ;
101                                    .ENABL  LC
102                                    .NLIST  BEX
103
104 000016  NTLER$ ,E1,8,CERR,RTSPC,,<Illegal event class>
105 000052  NTLER$ ,E2,8,CERR,RTSPC,,<Illegal event type mask word>
106 000120  NTLER$ ,E3,8,CERR,RTSPC,,<Event filter control block allocation failure>
107 000206  NTLER$ ,E4,8,CERR,RTSPC,,<Too many event type mask words>
108
109                                    .LIST   BEX
110                                    .DSABL  LC
111                                    .EVEN
112 000000  .PSECT
```

```

131
132
133
134
135
136
137
138
139
140
141
142
143
144
145 000116 121021
146 000120 001015
147 000122 005004
148 000124 112004
149 000126 001002
150
151 000130 005305
152 000132 000771
153
154 000134 122021
155 000136 001006
156
157 000140 005304
158 000142 001374
159
160 000144 005305
161 000146 001363
162 000150 000241
163 000152 000401
164
165 000154 000261
166 000156
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182 000160 010046
183
184 000162 105020
185 000164 005301
186 000166 001375
187 000170 012600

      .SBTTL  CMPSTR  -  COMPARE FILE NAME BLOCK FIELDS

      ;+
      INPUTS  -
                  R0,R1 ADDRESS OF FILE BLOCKS TO BE MATCHED
      OUTPUTS -
                  C CLEAR, MATCH
                  C SET,  NO MATCH
      ;--

CMPSTR: CMPB(R0),(R1)+      ; MATCH ON FIELD SIZE
        BNE 15$            ; IF NOT - BRANCH
5$:     CLR  R4            ; INIT BYTE COUNT
        MOVB (R0)+,R4      ; GET FILED SIZE
        BNE 10$           ; IF FILED LENGTH - BRANCH

        DEC  R5            ; COUNT ONE LESS FIELD
        BR  CMPSTR        ; CHECK NEXT FIELD

10$:    CMPB (R0)+,(R1)+   ; MATCH ?
        BNE 15$           ; IF NOT - BRANCH

        DEC  R4            ; COUNT ONE LESS CHARACTER
        BNE 10$           ; IF MORE - BRANCH

        DEC  R5            ; COUNT ONE LESS FIELD
        BNE  CMPSTR       ; IF MORE BRANCH
        CLC                ; INDICATE BLOCK MATCH
        BR  20$           ; CONTINUE

15$:    SEC                ; INDICATE NO MATCH
20$:    RETURN

      ;+
      INIBUF - INITIALIZE BUFFER
      ;--
      INPUTS  -
                  R0 - BUFFER ADDRESS
                  R1 - BYTE COUNT
      OUTPUTS -
                  BUFFER INIT
                  R0 PRESERVED
      ;--

INIBUF: MOV  R0,-(SP)      ; SAVE BUFFER ADDRESS

10$:     CLRB (R0)+        ; CLEAR BYTE
        DEC  R1            ; COUNT ONE LESS BYTE
        BNE 10$           ; IF MORE - BRANCH
        MOV  (SP)+,R0      ; RESTORE BUFFER ADDRESS
  
```

CF2AC5      CREATED BY    MACRO    ON 29-JUN-85 AT 00:11      PAGE 4      M 8

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|---|
| Z.PCB   | 000012     | #6-69   |
| Z.SCH   | 000007     | #6-69   |
| \$CEACH | ***** GX   | 7-114      11-285      11-306      12-369      12-374 |
| \$ERRFS | 000000 R   | #6-80      12-363                                     |
| \$HEADR | = ***** GX | 7-109      11-278                                     |
| \$LINKX | = ***** GX | 11-313  |
| \$XALOC | = ***** GX | 11-275  |
| .FLBK0  | 000502 RG  | #12-337   |
| .FLBK1  | 000520 RG  | #12-341   |
| .FLBK2  | 000536 RG  | #12-345   |
| .FLBK3  | 000554 RG  | #12-349   |
| .FLBK4  | 000572 RG  | #12-353   |

CF2  
ERR

```

359 001122          60$: RETURN
360
361          ;
362          ; STATION FLAGS
363          ;
364 001124 016700 000000G X.FLG:: MOV $SLTA,R0          ; GET SLT ADDRESS
365 001130 001435          BEQ 101$          ; IF EQ, NO PREVIOUS SLT$DF !!
366 001132 105760 000014          TSTB L.NSTA(R0)          ; IS THIS LINE A SLAVE ?
367 001136 001431          BEQ 10$          ; IF EQ, YES ... NO FLAGS BYTE
368 001140 016701 000000G          MOV STAN,R1          ; GET THE STATION NUMBER
369 001144 005267 000000G          INC STAN          ; SET FOR NEXT STATION
370 001150 006301          ASL R1          ; CONVERT STATION NUMBER TO 2 WORD INDEX
371 001152 006301          ASL R1
372 001154 060001          ADD R0,R1          ; GET START OF STATION TABLES
373 001156 062701 000022          ADD #L.MPF,R1
374 001162 142761 000100 000000          BICB #SF.ENA,S.FLG(R1)          ; CLEAR SETTABLE BITS
375 001170 142761 000040 000000          BICB #SF.MFL,S.FLG(R1)
376 001176 156761 000000G 000000          BISB MRKFL,S.FLG(R1)          ; SET FLAG BITS
377 001204 032760 004000 000000          BIT #LF.MFL,L.FLG(R0)          ; IS THE LINE MARKED FOR LOAD ?
378 001212 001003          RNE 10$          ; IF NE, YES ... OKAY
379 001214 142761 000100 000000          BICB #SF.ENA,S.FLG(R1)          ; ELSE, MAKE SURE WE DON'T ENABLE LINE
380 001222          10$: RETURN
381          ;
382          ; ERROR CONDITIONS
383          ;
384 001224          101$: MSG$R YM          ; STATION NOT IN SYSTEM

```

```

188 000440 001004      BNE      8$          ; IF NE, YES
189 000442 012722 063147 MOV      #*RP00,(R2)+ ; ELSE DEFAULT TO 'POOL..'
190 000446 012712 047574 MOV      #*RL..(R2)
191 000452 012702 000000G 8$: MOV      #,PNAM2,R2 ; POINT AT SECOND NAME
192 000456          10$: SWSTK$ 70$ ; * ENTER KERNEL MODE
193 000462 016701 000000G MOV      PARHD,R1 ; * POINT AT PARTITION LIST
194 000466 011101      20$: MOV      (R1),R1 ; * GET NEXT PCB IN LIST
195 000470 001004      BNE      30$ ; * IF NE, NOT AT END OF LIST
196 000472 012766 000000G 000002 MOV      #SERRXC,2(SP) ; * ERROR - PARTITION NOT FOUND
197 000500 000436      BR      60$ ; *
198 000502 026112 000000G 30$: CMP      P,N'"(R1),(R2) ; * CHECK FOR RIGHT NAME
199 000506 001367      BNE      20$ ; * TWO WORDS
200 000510 026162 000002G 000002 CMP      P,NAM+2(R1),2(R2) ; *
201 000516 001363      BNE      20$ ; *
202
203          .IF NDF R$$MPL
204
205 000520 032761 000000G 000000G BIT      #PS.SYS,P.STAT(R1) ; * IS IT SYSTEM-CONTROLLED ?
206 000526 001004      BNE      40$ ; * IF NE, YES
207 000530 012766 000000G 000002 MOV      #SERRXE,2(SP) ; * ERROR - NOT SYSTEM-CONTROLLED PARTITION
208 000536 000417      BR      60$ ; *
209
210          .ENOC
211
212 000540 010105      40$: MOV      R1,R5 ; * COPY MAIN PCB ADDRESS
213 000542 016701 000000G MOV      PLGTH,R1 ; * SUB-PCB LENGTH
214 000546          CALL      @ALOCB ; * TRY TO ALLOCATE
215 000552 103004      BCC      50$ ; * IF CC, SUCCESS
216 000554 012766 000000G 000002 MOV      #SERRXJ,2(SP) ; * ERROR - SUB-PCB ALLOCATION FAILURE
217 000562 000405      BR      60$ ; *
218 000564          50$: CALL      CRSUB ; * CREATE SUB-PARTITION
219 000570 103402      BCS      60$ ; * IF CS, ERROR
220 000572 010566 000004 MOV      R5,4(SP) ; * RETURN SUB-PCB ADDRESS IN USER R1
221 000576          RETURN ; * BACK TO USER MODE
222 000600 005700      70$: TST      R0 ; * DID ERROR OCCUR IN KERNEL MODE ?
223 000602 001401      BEQ      80$ ; * IF EQ, NO
224 000604          CALL      @ (R0)+ ; * GO PRINT ERROR MESSAGE
225 000606 010167 000000G 80$: MOV      R1,.PCB ; * WAS SUB-PCB CREATED ?
226 000612 001003      BNE      90$ ; * IF NE, YES
227 000614          EMG$R      XK ; * PARTITION TOO FRAGMENTED
228
229 000622          90$: .IF DF R$$MPL
230
231          TSTB      @NCPU ; * IS THIS A MULTI-PROCESSOR SYSTEM ?
232          BEQ      93$ ; * IF EQ, NO
233          MOV      R3,-(SP) ; * SAVE R3
234          MOV      @NCPU,R3 ; * ASSUME AT MOST 177 CPU'S
235          91$: MOV      R3,R0 ; * COPY CPU NUMBER
236          ASL      R0 ; * CONVERT TO A WORD INDEX
237          MOV      K6TAB,K6TMP ; * STORE $K6TAB VALUE
238          ADD      R0,K6TMP ; * GET ADDRESS OF BIAS FOR THIS CPU
239          SUB      #2,K6TMP ; *
240          MOV      @K6TMP,R0 ; * GET BIAS
241          CALL      $UMRAL ; * ALLOCATE UMR'S FOR R3'S CPU
242
243          92$: SOB      R3,91$ ; * REPEAT <CPU NUMBER> TIMES
244

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43

.TITLE CF2AC8 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1983, 1984, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 4.00 7-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

CF2AC8      CREATED BY    MACRO    ON 29-JUN-85 AT 00:12      PAGE 2      M 12  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE    | REFERENCES   |
|---------|----------|--|
| L.COST  | 000015   | #5-62  |
| L.CTL   | 000012   | #5-62      11-403                                    |
| L.CVA   | 177776   | #5-62  |
| L.DDM   | 000002   | #5-62      11-401                                    |
| L.DDS   | 000004   | #5-62  |
| L.DLC   | 000003   | #5-62  |
| L.DLM   | 000006   | #5-62  |
| L.DLS   | 000010   | #5-62  |
| L.FLG   | 000000   | #5-62  |
| L.KRBA  | 000016   | #5-62  |
| L.LEN   | = 000022 | #5-62  |
| L.MPF   | = 000022 | #5-62  |
| L.NMST  | 000020   | #5-62  |
| L.NSTA  | 000014   | #5-62  |
| L.OWNR  | 000021   | #5-62  |
| L.UNT   | 000013   | #5-62      11-405                                    |
| N\$SVCT | = *****  | 8-191  |
| PAS     | = 000460 | RG #9-265  |
| PDVNM   | = *****  | GX 10-352  |
| PDVTA   | = *****  | GX 10-351      10-363                                |
| RTSPC   | = *****  | GX 5-85  |
| R\$S11D | = *****  | 5-61   |
| R\$S11M | = 000000 | 5-61   |
| R\$S11S | = *****  | 5-61   |
| SERADD  | = *****  | GX 8-158      8-213      8-214      8-215            |
| SERALL  | = 000162 | R 8-178      #8-186                                  |
| SERBEG  | = *****  | GX 7-135   |
| SERCR   | = 000632 | RG #10-328   |
| SERCUR  | = *****  | GX *8-196  |
| SERDCT  | = *****  | GX 8-226   |
| SERDEV  | = *****  | GX 8-218   |
| SERDMP  | = *****  | GX 8-224      8-225                                  |
| SEREND  | = 000144 | RG #8-178  |
| SERFIN  | = *****  | GX 7-137   |
| SERFLG  | = *****  | GX 8-209   |
| SERHST  | = *****  | GX 8-211   |
| SERINI  | = 000056 | RG #7-135  |
| SERNOD  | = *****  | GX 8-210   |
| SERPAS  | = *****  | GX 8-219      8-220      8-222      8-223      9-282 |
| SERSLN  | = 000000 | RG #6-105  |
| SERSLT  | = *****  | GX *6-118      *6-119      8-216      *11-408        |
| SERTMP  | = *****  | GX 8-170   |
| SF.ACT  | = 000200 | #5-62  |
| SF.CIR  | = 000010 | #5-60  |
| SF.DEV  | = 000020 | #5-60  |
| SF.DIA  | = 004000 | #5-60  |
| SF.DPA  | = 000100 | #5-60  |
| SF.DPC  | = 000200 | #5-60  |
| SF.DUM  | = 010000 | #5-60  |
| SF.ENA  | = 000100 | #5-62  |
| SF.HAD  | = 000004 | #5-60  |
| SF.HST  | = 000002 | #5-60  |



```

263
264
265      + FILE - STORE FILE NAME IN FILBUF
266
267      INPUTS -
268          STRADD - CURRENT STRING ADDRESS
269          STRSZ - STRING SIZE
270
271      OUTPUTS -
272          FILE NAME STORED IN FILBUF
273      -
274
275      000510 012702 000000G FILE:: MOV #FILBUF,R2 ; GET FILE NAME BUFFER ADDRESS
276      000514 116722 000000G      MOVB STRSZ,(R2)+ ; STORE LENGTH OF STRING
277      000520 016700 000000G      MOV STRADD,R0 ; GET ADDRESS OF STRING
278
279      000524 112022 000000G 1$: MOVB (R0)+(R2)+ ; COPY STRING INTO BUFFER
280      000526 005367 000000G      DEC STRSZ ; COUNT ONE LESS CHARACTER
281      000532 001374 000000G      BNE 1$ ; IF MORE - BRANCH
282      000534 000000G 000000G      MOV #FNBUFF,STRPT ; RESET BUFFER ADDRESS
283      000542 000000G      CLR STRSZ ; INIT STRING SIZE
284
285      000546      RETURN
286
287      + EXT - STORE FILE EXTENSION
288
289      INPUTS -
290          STRADD - STRING ADDRESS
291          STRSZ - STRING SIZE
292
293      OUTPUTS -
294          FILBUF CONTAINS FILE EXTENSION
295      -
296
297      000550 022767 000003 000000G EXT:: CMP #EXTN,STRSZ ; CHECK FOR LEGAL SIZE
298      000556 100432 000000G      BMI 10$ ; IF TOO LARGE - BRANCH
299      000560 016700 000000G      MOV STRADD,R0 ; GET STRING ADDRESS
300      000564 005002 000000G      CLR R2 ; INIT REG FOR FILENAME SIZE
301      000566 116702 000000G      MOVB FILBUF,R2 ; GET SIZE OF FILE NAME
302      000572 012701 000000G      MOV #FILBUF,R1 ; GET FILE NAME BUFFER ADDRESS
303      000576 005201 000000G      INC R1 ; SKIP SIZE BYTE
304      000600 060102 000000G      ADD R1,R2 ; CALCULATE STARTING ADDR. OF EXTENSION
305      000602 112722 000056 000000G      MOVB #1,(R2)+ ; INSERT DOT
306      000606 005267 000000G      INC FILBUF ; BUMP BYTE COUNT
307      000612 066767 000000G 000000G      ADD STRSZ,FILBUF ; ADJUST FILE NAME SIZE
308      000620 112022 000000G 5$: MOVB (R0)+(R2)+ ; COPY FILE EXTENSION
309      000622 005367 000000G      DEC STRSZ ; COUNT ONE LESS BYTE
310      000626 001374 000000G      BNE 5$ ; IF MORE - BRANCH
311
312      000630 012767 000000G 000000G      MOV #FNBUFF,STRPT ; RESET BUFFER POINTER
313      000636 005067 000000G      CLR STRSZ ; INIT STRING SIZE
314
315      000642      RETURN
316
317      000644 062716 000002 10$: ADD #2,(SP) ; SYNTAX ERROR
318      000650      RETURN
319

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL  | VALUE    | REFERENCES           |
|---------|----------|----------------------|
| ZF.DDM  | = 000001 | #6-65                |
| ZF.DIA  | = 004000 | #6-65                |
| ZF.DLC  | = 000002 | #6-65                |
| ZF.DVP  | = 100000 | #6-65                |
| ZF.INI  | = 040000 | #6-65                |
| ZF.KMX  | = 000020 | #6-65                |
| ZF.LLC  | = 000004 | #6-65                |
| ZF.LMC  | = 000100 | #6-65                |
| ZF.MAN  | = 020000 | #6-65                |
| ZF.MFL  | = 000010 | #6-65                |
| ZF.MTM  | = 000400 | #6-65                |
| ZF.MUX  | = 000040 | #6-65                |
| ZF.PSE  | = 002000 | #6-65                |
| ZF.SLI  | = 010000 | #6-65                |
| ZF.TIM  | = 000200 | #6-65                |
| ZF.X3P  | = 000000 | #6-65                |
| ZS.ASN  | = 100000 | #6-65                |
| ZS.BSY  | = 140000 | #6-65                |
| Z.AVL   | 000014   | #6-65                |
| Z.DAT   | 000016   | #6-65                |
| Z.DSP   | 000000   | #6-65 6-65           |
| Z.FLG   | 000010   | #6-65                |
| Z.LEN   | = 000016 | #6-65                |
| Z.LLN   | 000006   | #6-65                |
| Z.MAP   | 000020   | #6-65                |
| Z.NAM   | 000004   | #6-65                |
| Z.PCB   | 000012   | #6-65                |
| Z.SCH   | 000007   | #6-65                |
| \$ALPHA | = 000022 | #10-398              |
| \$ANY   | = 000020 | #10-398              |
| \$BLANK | = 000006 | #10-398              |
| \$CAT5  | = *****  | GX 7-204 7-207 9-362 |
| \$DIGIT | = 000024 | #10-398              |
| \$DNUMB | = 000014 | #10-398              |
| \$EOS   | = 000012 | #10-398              |
| \$ERRXA | = *****  | GX 7-140             |
| \$ERRXB | = *****  | GX 7-147             |
| \$ERRXD | = *****  | GX 7-172             |
| \$ERRXF | = *****  | GX 7-252             |
| \$ERR1T | = *****  | GX 7-103             |
| \$EXIT  | = 000000 | #10-398              |
| \$FAIL  | = 177777 | #10-398              |
| \$GPRM  | = *****  | 10-398               |
| \$LAMD  | = 000000 | #10-398              |
| \$NUMBR | = 000002 | #10-398              |
| \$QBRA  | 000000   | RG #7-84             |
| \$QMLMC | 000014   | RG #7-90             |
| \$QXPT  | 000006   | RG #7-87             |
| \$RAD50 | = 000016 | #10-398              |
| \$RONLY | = *****  | 10-398 10-398        |
| \$STRNG | = 000004 | #10-398              |
| \$SUBXP | = 000010 | #10-398              |

CF2DEC CREATED BY MACRO ON 29-JUN-85 AT 00:14 PAGE 1 M 15

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL   | VALUE      | REFERENCES |
|----------|------------|------------|
| CFGBF    | = ***** GX | 6-66       |
| CFGSZ    | = ***** GX | 6-65       |
| DECDF    | 000000 R   | 6-62       |
| DECKW    | 000000 RG  | 6-64       |
| DECP     | = ***** GX | 7-86       |
|          |            | 7-162      |
| DECST    | 000000 PG  | #8-177     |
| DELF     | 000256 RG  | #7-126     |
| DELW     | 000306 RG  | #7-134     |
| D\$DELF  | 000045     | *7-129     |
| D\$DELW  | 000046     | *7-137     |
| D\$INAC  | 000044     | *7-121     |
| D\$INCT  | 000042     | *7-105     |
| D\$IPL   | 000051     | *7-145     |
| D\$LST   | 000047     | *7-154     |
| D\$MLL   | 000040     | *7-87      |
| D\$NOD   | 000041     | *7-96      |
| D\$OUTT  | 000043     | *7-113     |
| D\$RETF  | 000050     | *7-163     |
| D\$SEG   | 000036     | *7-169     |
| ECLLNK   | 000052 RG  | #7-84      |
| INAC     | 000226 RG  | #7-118     |
| INCT     | 000146 RG  | #7-102     |
| IPL      | 000336 RG  | #7-142     |
| LST      | 000366 RG  | #7-151     |
| NODCT    | 000110 RG  | #7-93      |
| OUTT     | 000176 RG  | #7-110     |
| RETF     | 000416 RG  | #7-160     |
| SEG      | 000446 RG  | #7-168     |
| SYNERR   | = ***** GX | *6-70      |
| \$ALPHA  | = 000022   | #8-177     |
| \$ANY    | = 000020   | #8-177     |
| \$BLANK  | = 000006   | #8-177     |
| \$DIGIT  | = 000024   | #8-177     |
| \$DNUMB  | = 000014   | #8-177     |
| \$EOS    | = 000012   | #8-177     |
| \$ERRXO  | = ***** GX | 7-91       |
| \$ERRXP  | = ***** GX | 7-100      |
| \$ERRXO  | = ***** GX | 7-132      |
| \$ERRXR  | = ***** GX | 7-140      |
| \$ERRXS  | = ***** GX | 7-148      |
| \$ERRXT  | = ***** GX | 7-157      |
| \$ERRXU  | = ***** GX | 7-108      |
| \$ERRXW  | = ***** GX | 7-124      |
| \$ERRXX  | = ***** GX | 7-166      |
| \$ERRXY  | = ***** GX | 7-116      |
| \$ERRIT  | = ***** GX | 6-74       |
| \$EXIT   | = 000000   | #8-177     |
| \$FAIL   | = 177777   | #8-177     |
| \$GPRM   | = *****    | 8-177      |
| \$LAMBDA | = 000000   | #8-177     |
| \$NUMBR  | = 000002   | #8-177     |

CF2UMR      CREATED BY    MACRO    ON 29-JUN-85 AT 00:14      PAGE 2      M 16

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |
|------------|------------|
| CALL       | 5-90       |
| CALLR      | 5-167      |
| EMSG\$R    | #5-58      |
| NHWD\$     | #5-62      |
| RETURN     | 5-95       |
| SWSTK\$    | 5-90       |
|            | 5-92       |
|            | 5-112      |
|            | 5-128      |
|            | 5-166      |
|            | 6-183      |
|            | 5-156      |
|            | 5-168      |
|            | 5-64       |
|            | 5-134      |
|            | 6-195      |
|            | 5-166      |
|            | 6-183      |

CFGX3P - CONFIG FILE SCAN ACTION MACRO V05.03b Saturday 29-Jun-85 <sup>N 1</sup> 00:09  
Table of contents

|     |     |                                 |
|-----|-----|---------------------------------|
| 5-  | 55  | MACRO DEFINITIONS               |
| 6-  | 62  | CONSTANT DEFINITIONS            |
| 7-  | 72  | LOCAL DATA                      |
| 8-  | 85  | ERROR MESSAGES                  |
| 9-  | 100 | \$QX3P - LOOK FOR X3P\$DF MACRO |
| 10- | 129 | X3P\$DF STATE TABLE             |
| 11- | 176 | X3P\$DF ACTION ROUTINES         |

CFGX  
MACR  
MACR  
CALL  
DBGT  
EMSG  
ISTA  
MTRAI  
NTLEI  
RETU  
STAT  
TRAN

## MACRO CROSS REFERENCE

CREF    04.00

## MACRO NAME      REFERENCES

|         |         |         |         |         |         |         |         |         |         |         |  |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|--|
| CALL    | 9-119   | 11-181  | 11-189  | 11-200  | 11-208  |         |         |         |         |         |  |
| DBGTP\$ | #10-133 | #10-169 |         |         |         |         |         |         |         |         |  |
| EMSG\$R | #5-60   | 9-127   | 11-185  | 11-195  | 11-196  | 11-204  | 11-214  | 11-215  | 11-237  |         |  |
| ISTAT\$ | #5-60   | 10-133  |         |         |         |         |         |         |         |         |  |
| MTRAN\$ | #10-133 |         |         |         |         |         |         |         |         |         |  |
| NTLER\$ | #5-60   | 8-91    | 8-92    | 8-93    | 8-94    | 8-95    |         |         |         |         |  |
| RETURN  | 9-123   | 11-184  | 11-194  | 11-203  | 11-213  | 11-233  | 11-236  | 11-244  |         |         |  |
| STAT\$  | #5-60   | 10-137  | #10-140 | #10-143 | #10-146 | #10-149 | #10-152 | #10-155 | #10-158 | #10-161 |  |
|         | #10-164 | #10-167 | #10-171 | #10-174 |         |         |         |         |         |         |  |
| TRANS   | #5-60   | #10-138 | #10-141 | #10-144 | #10-147 | #10-150 | #10-153 | #10-156 | #10-159 | #10-162 |  |
|         | #10-165 | #10-168 | #10-169 | #10-172 |         |         |         |         |         |         |  |

```

346 .SBTTL QUICK - FAST REMOTE NODE LIST PROCESSING
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362 000766 016700 000000G QUICK: MOV $LSTRM,R0 ; GET LAST REMOTE NODE POINTER
363 000772 001002 BNE 10$ ; IF VALID - BRANCH
364 000774 000261 5$: SEC ; INDICATE FAILURE
365 000776 RETURN
366
367 001000 010001 10$: MOV R0,R1 ; SAVE LAST REMOTE LIST ADDRESS
368 001002 026767 000006' 000000G CMP RADDR,$LSTAD ; DOES BLOCK GO AT END OF LIST
369 001010 103771 BLO 5$ ; IF NO - BRANCH
370 001012 000241 CLC ; IF YES - INDICATE SUCCESS
371 001014 010267 000000G MOV R2,$LSTRM ; UPDATE NEW END OF LIST POINTER
372 001020 016767 000006' 000000G MOV RADDR,$LSTAD ; UPDATE HIGHEST NODE ADDRESS
373
374 001026 RETURN
375
376 000001 .END
  
```

225  
226  
227  
228

.DSABL LSB

000001

.END



```

259
260
261
262          ;+
263          ; ROUSDF ACTION ROUTINES
264          ; -
265 000452 005767 000000G R.BEA:: TST .NN          ; ROUTING NODE ?
266 000456 001004          BNE 10$          ; IF YES - BRANCH
267 000460 005767 000000G TST .PNUMB          ; IF NO - THIS VALUE MUST BE ZERO
268 000464 001405          BEQ 20$          ; IF LEGAL - BRANCH
269 000466 000410          BR 101$          ; ELSE - REPORT ERROR
270
271 000470 022767 002000 000000G 10$: CMP #1024...PNUMB ; CHECK AGAINST HIGH LIMIT
272 000476 103404          BLD 101$          ; IF OUT OF RANGE - BRANCH
273
274 000500 016767 000000G 000000G 20$: MOV .PNUMB,.NBEA ; COPY NUMBER OF BROADCAST ENDNODE ADJACENCIES
275 000506          RETURN
276
277 000510          101$: MSG$R Y0          ; ILLEGAL BROADCAST ENDNODE ADJACENCY PARAMETER
278
279 000516 022767 000036 000000G R.AMXH::CMP #30...PNUMB ; CHECK AGAINST HIGH LIMIT
280 000524 103404          BLD 101$          ; IF OUT OF RANGE - BRANCH
281
282 000526 016767 000000G 000000G MOV .PNUMB,.AMAXH ; COPY VALUE OF AREA MAX HOPS
283 000534          RETURN
284
285 000536          101$: MSG$R YQ          ; ILLEGAL AREA MAX HOP PARAMETER
286
287
288 000544 022767 002000 000000G R.AMXC::CMP #1024...PNUMB ; CHECK AGAINST HIGH LIMIT
289 000552 103404          BLD 101$          ; IF OUT OF RANGE - BRANCH
290
291 000554 016767 000000G 000000G 20$: MOV .PNUMB,.AMAXC ; COPY AREA MAX COST PARAMETER
292 000562          RETURN
293
294 000564          101$: MSG$R YP          ; ILLEGAL AREA MAX COST VALUE

```

```

114                                     .SBTTL  FLTDF$ ACTION ROUTINES FOR EVENT CLASS AND TYPE
115
116
117                                     ;+
118                                     ; E.NUM - PROCESS EVENT CLASS NUMBER
119                                     ;+
120                                     ;+
121                                     ;+
122                                     ;+
123 000000 005767 000000G  E.NUM:: TST      .PNUMH      ; SPECIFIED VALUE A WORD VALUE?
124 000004 001022          BNE      101$      ; BR IF NO
125 000006 016700 000000G  MOV      .PNUMB,R0    ; GET SPECIFIED VALUE
126 000012 022700 077700  CMP      #77700,R0    ; LEGAL VALUE?
127 000016 103415          BLO      101$      ; BR IF NO
128 000020 010067 000000'  MOV      R0,CLASS    ; SAVE CLASS
129 000024 005067 000002'  CLR      FLAG      ; START OFF WITH NO DESTINATIONS DEFINED
130 000030 012700 000004'  MOV      #EVENT,R0   ; GET ADDRESS OF EVENT TYPE MASKS
131 000034 010067 000014'  MOV      R0,STRADD    ; SAVE IT
132          000004          .REPT      4
133          .CLR      (R0)+      ; CLEAR EVENT TYPE MASKS
134          .ENDR
135 000050          RETURN
136
137                                     ; ERROR CONDITIONS
138
139 000052          101$:  MSG$R  E1      ; ILLEGAL EVENT CLASS
140
141                                     ;+
142                                     ;+
143                                     ; E.MSK - PROCESS EVENT TYPE MASK
144                                     ;+
145                                     ;+
146                                     ;+
147 000060 005767 000000G  E.MSK:: TST      .PNUMH      ; LEGAL MASK?
148 000064 001013          BNE      101$      ; BR IF NO
149 000066 022767 000014' 000014'  CMP      #EVENT+10,STRADD ; MORE THAN 4 EVENT TYPE MASK WORDS?
150 000074 001412          BEQ      102$      ; BR IF YES
151 000076 016777 000000G 000014'  MOV      .PNUMB,@STRADD ; SAVE THE MASK
152 000104 062767 000002 000014'  ADD      #2,STRADD    ; POINT TO NEXT WORD IN AREA
153 000112          RETURN
154
155                                     ; ERROR CONDITIONS
156
157                                     ;+
158 000114          101$:  MSG$R  E2      ; ILLEGAL MASK WORD
159 000122          102$:  MSG$R  E4      ; TOO MANY EVENT TYPE MASK WORDS

```

CF2AC5 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 <sup>N 7</sup> 00:11 Page 8-1  
CMPSTR - COMPARE FILE NAME BLOCK FIELDS

188  
189 000172

RETURN

CF2  
MAC  
MAC  
CAL  
DHB  
EMS  
FNB  
NTL  
PDV  
RET  
RET  
SER  
SLT  
SWS

CF2AC5      CREATED BY    MACRO    ON 29-JUN-85 AT 00:11      PAGE 5      N 8

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES  |
|------------|---|
| CALL       | 7-101      7-114      7-122      10-233      10-243      10-247      10-252      11-273      11-275      11-285<br>11-306      11-313      12-356      12-357      12-360      12-366      12-369      12-374      12-391      12-395<br>12-399 |
| DHBD\$     | #5-58      6-66   |
| EMSG\$R    | #5-58      12-363   |
| FNBD\$     | #5-58      6-67   |
| NTLR\$     | #5-58      6-80   |
| PDVDF\$    | #5-59      6-69   |
| RETC       | #5-58      7-109      11-278  |
| RETURN     | 7-128      8-166      8-189      9-214      10-257      11-316      12-382      12-401  |
| SERDF\$    | #5-58      6-68   |
| SLTDF\$    | #5-59      6-70   |
| SWSTK\$    | 7-101      11-273      12-366   |

```

386                                     :
387                                     : L.LGST - SET LOGGING STATE ACTION ROUTINE (LOG$ST)
388                                     :
389 001232 105767 000001G L.LGST::TSTB ,PNUMB+1 ;IS SPECIFIED VALUE A BYTE VALUE?
390 001236 001011 BNE 101$ ;BR IF NO - ERROR
391 001240 126727 000000G 000007 CMPB ,PNUMB,#7 ;LEGAL VALUES ARE 0-7
392 001246 101005 BHI 101$ ;BR IF ILLEGAL VALUE
393 001250 116777 000000G 000000G MOVB ,PNUMB,@LGSTT ;SAVE IT
394 001256 002401 BLT 101$ ;BR IF ILLEGAL VALUE
395 001260 RETURN
396 001262 101$: MSG$R L1
397
398 000001 .END
  
```

```

245      MOV      (SP)+,R3      ; RESTORE R3
246      CLR      R2           ; ZERO THE UMR BLOCK ADDRESS
247      BR       94$         ; AND CONTINUE
248
249      .ENDC
250
251      000622      93$:      .IF DF R$$MPL
252                        BIT    #HF.UBM,@HFMSK ; DOES THIS SYSTEM HAVE UMR'S?
253                        BNE    931$          ; BR IF YES
254                        INC     .UMRFL       ; ELSE INDICATE NO UMRS
255
256      931$:      .IFF
257
258      000622      032777      000000G 000000G      BIT    #FE.EXT,@FMASK ; Extended memory system?
259      000630      001403      BEQ    931$          ; Br if no
260      000632      005767      000000G      TST    $UMRPT    ; Are UMR's present?
261      000636      001002      BNE    933$          ; Br if yes
262      000640      005267      000000G      931$:    INC     .UMRFL       ; Indicate no UMRS
263      000644      933$:
264
265      .ENDC
266
267      000644      032777      000000G 000000G 94$:    CALL    $UMRAL      ; ALLOCATE AND LOAD UMR'S
268      000650      001403      BIT    #FE.PLA,@FMASK ; IS THIS A PLAS SYSTEM ?
269      000656      001403      BEQ    95$          ; NO
270      000660      012761      177774      000000G      MOV    #C<3>,P.PROC(R1) ; YES... ALLOW ONLY SYSTEM READ/WRITE ACCESS
271      000666      005061      000000G      CLR    P.ATT(R1) ; AND INITIALIZE ATTACHMENT LISTHEAD
272      000672      010161      000002G      MOV    R1,P.ATT+2(R1)
273      000676      062761      000000G 000002G      ADD    #P.ATT,P.ATT+2(R1)
274      000704      016100      000000G      95$:    MOV    P.REL(R1),R0 ; GET HOME BLOCK BIAS
275      000710      010067      000000G      MOV    R0,HOME ; SAVE IT
276      000714      SWSTK$      100$          * ENTER KERNEL MODE
277      000720      CALL    $MAPX ; * MAP TO THE HOME BLOCK
278                                * REMEMBER - $MAPX ADDS 1 WORD TO STACK
279      000724      016622      000006      MOV    6(SP),(R2)+ ; * STORE NTPool PCB ADDRESS
280      000730      016612      000010      MOV    10(SP),(R2) ; * AND UMR BLOCK ADDRESS
281      000734      100$:      RETURN ; * BACK TO USER MODE AND THEN TO CALLER
282
283      .IFF ; DF R$$11M
284
285      MOV    #.PNAM1,R2 ; POINT TO REGION NAME
286      TST    (R2) ; WAS A REGION NAME SPECIFIED?
287      BNE    10$ ; IF NE, YES
288      MOV    #*RPOD,(R2) ; USE DEFAULT NAME
289      MOV    #*RL,,2(R2) ;
290      10$:    MOV    .NTPSZ,,POSIZ ; SET SIZE OF BUFFER AREA
291      ADD    .FUDGE,,POSIZ ; ADD IN FUDGE FACTOR TO POOL SIZE
292      BIT    #LS.TOP,$FUNC ; TOP-DOWN LOADING REQUIRED?
293      BEQ    15$ ; BR IF NO
294      BIS    #RS.HLD,.PORDB+R.GSTS ; SET TOP-DOWN LOADING FLAG
295      15$:    CRRG$S      #.PORDB ; CREATE REGION FOR NETWORK POOL
296      BCS    101$ ; IF CS, FAILURE
297      MOV    .RGID,.WRGID ; SET UP REGION ID FOR MAPPING
298      MAP$S    #WDB ; MAP TO THE REGION
299      BCS    102$ ; IF CS, FAILURE
300      MOV    R3,-(SP) ; GET 2 FREE REGISTERS
301      MOV    R4,-(SP) ;

```

```

45          .SBTTL  MACRO DEFINITIONS
46
47          ;
48          ; LOCAL MACROS
49          ;
50          .MCALL  EMSG$,NTLR$,RETC,FNBDF$,DHBDF$,SERDF$
51          .MCALL  PDVDF$,SLTDF$
52
53          ;
54          ;
55          ;+
56          ; OFFSET VALUES
57          ; -
58          000000          DHBDF$          ; DEFINE DECNET HOME BLOCK OFFSETS
59          000000          FNBDF$          ; DEFINE FILE NAME BLOCK OFFSETS
60          000000          SERDF$          ; DEFINE SERVICE BLOCK OFFSETS
61          000000          PDVDF$          ; DEFINE PDV OFFSETS
62          000000          SLTDF$          ; DEFINE SLT OFFSETS
63
64          ;
65          .MACRO  SAVMAP
66          MOV     @KSAR5,-(SP)
67          .ENDM
68
69          .MACRO  RESMAP
70          MOV     (SP)+,@KSAR5
71          .ENDM
72
73          ;
74          ;+
75          ; LOCAL EQUIVS
76          ; -
77
78          000100          LET      = 100          ; HEX LETTER DIGIT
79
80          ;
81          ;+
82          ; LOCAL ERROR MESSAGES
83          ; -
84          000000          .PSECT  DATA
85          000000          NTLR$      ,SS,8,CERR,RTSPC,CFLIN,<Service block allocation failure>
86
87          000000          .PSECT
88

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL    | VALUE    | REFERENCES                        |
|-----------|----------|-----------------------------------|
| SF.LOA    | = 002000 | #5-60                             |
| SF.LPB    | = 000004 | #5-62                             |
| SF.MFL    | = 000040 | #5-62                             |
| SF.PAC    | = 000020 | #5-62                             |
| SF.PH3    | = 020000 | #5-60                             |
| SF.PSS    | = 000040 | #5-60                             |
| SF.REA    | = 000010 | #5-62                             |
| SF.SEC    | = 000400 | #5-60                             |
| SF.SER    | = 000001 | #5-62                             |
| SF.SVC    | = 000002 | #5-62                             |
| SF.TER    | = 001000 | #5-60                             |
| SF.UNL    | = 000040 | #5-62                             |
| SLTMA     | = *****  | GX 11-391                         |
| SLTNM     | = *****  | GX 11-396                         |
| S\$B\$BAS | = *****  | 5-85                              |
| S.ADD     | 000004   | #5-60 *8-210                      |
| S.CIR     | 000016   | #5-60 *8-216                      |
| S.COST    | 000001   | #5-62                             |
| S.DEV     | 000020   | #5-60 *8-218                      |
| S.DIA     | 000046   | #5-60                             |
| S.DPA     | 000032   | #5-60 *8-224 *8-225               |
| S.DPC     | 000036   | #5-60 *8-226                      |
| S.DUM     | 000050   | #5-60                             |
| S.FLAG    | 000002   | #5-60 *8-209                      |
| S.FLG     | 000000   | #5-62                             |
| S.HAD     | 000010   | #5-60 *8-213 *8-214 *8-215        |
| S.HST     | 000006   | #5-60 *8-211                      |
| S.LEN     | 000004   | #5-62                             |
| S.LENG    | 000052   | #5-60 8-187 8-203                 |
| S.LNK     | 000000   | #5-60                             |
| S.LOA     | 000044   | #5-60                             |
| S.NMST    | 000002   | #5-62                             |
| S.OWNR    | 000003   | #5-62                             |
| S.PSS     | 000022   | #5-60 *8-219 *8-220 *8-222 *8-223 |
| S.SEC     | 000040   | #5-60                             |
| S.TER     | 000042   | #5-60                             |
| X\$M\$CB  | = *****  | 5-61                              |
| ZF.COJ    | = 001000 | #5-61                             |
| ZF.DDM    | = 000001 | #5-61                             |
| ZF.DIA    | = 004000 | #5-61                             |
| ZF.DLC    | = 000002 | #5-61                             |
| ZF.DVP    | = 100000 | #5-61                             |
| ZF.INI    | = 040000 | #5-61                             |
| ZF.KMX    | = 000020 | #5-61                             |
| ZF.LLC    | = 000004 | #5-61                             |
| ZF.LMC    | = 000100 | #5-61                             |
| ZF.MAN    | = 020000 | #5-61                             |
| ZF.MFL    | = 000010 | #5-61                             |
| ZF.MTM    | = 000400 | #5-61                             |
| ZF.MUX    | = 000040 | #5-61                             |
| ZF.PSE    | = 002000 | #5-61                             |
| ZF.SLI    | = 010000 | #5-61                             |



CF2AC9 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:13 Page 8-1  
ACTION ROUTINES FOR PAR\$DF

320

C  
S  
S  
S  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.  
.

CF2AC9      CREATED BY    MACRO    ON 29-JUN-85 AT 00:13

PAGE 5      N 14

## SYMBOL CROSS REFERENCE

CREF    04.00

SYMBOL    VALUE      REFERENCES

|           |            |         |        |       |       |       |       |       |       |
|-----------|------------|---------|--------|-------|-------|-------|-------|-------|-------|
| \$\$\$FLG | = 177777   | #10-398 |        |       |       |       |       |       |       |
| \$\$\$KEY | = 177777   | #10-398 |        |       |       |       |       |       |       |
| .BRAD     | = ***** GX | *9-389  |        |       |       |       |       |       |       |
| .FUDGE    | = ***** GX | *7-216  | *7-218 | 7-235 |       |       |       |       |       |
| .NAMBF    | = ***** GX | 7-191   | 7-202  |       |       |       |       |       |       |
| .NOMAP    | = ***** GX | *7-226  | *7-228 | 7-247 |       |       |       |       |       |
| .NTPSZ    | = ***** GX | 7-168   | 7-234  | 7-249 |       |       |       |       |       |
| .PCB      | = ***** GX | *7-151  | 7-159  | 7-238 |       |       |       |       |       |
| .PNAM1    | = ***** GX | 7-117   |        |       |       |       |       |       |       |
| .PNAM2    | = ***** GX | 7-188   |        |       |       |       |       |       |       |
| .PNUMB    | = ***** GX | 7-216   | 7-226  | 9-334 | 9-337 | 9-340 | 9-343 | 9-349 | 9-352 |
|           |            | 9-367   | 9-372  | 9-380 | 9-383 | 9-386 | 9-389 |       | 9-358 |
| .PNUMH    | = ***** GX | 9-368   |        |       |       |       |       |       |       |
| .PSTCN    | = ***** GX | 7-196   |        |       |       |       |       |       |       |
| .PSTPT    | = ***** GX | 7-195   | 9-361  |       |       |       |       |       |       |
| .TFARS    | = ***** GX | 7-97    |        |       |       |       |       |       |       |
| .UMRFL    | = ***** GX | *7-258  |        |       |       |       |       |       |       |

PAGE 2 N 15

CREF 04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|---|
| \$QDEC  | = 000000 RG | #6-62   |
| \$RAD50 | = 000016    | #8-177  |
| \$RONLY | = *****     | #8-177      8-177      8-177  |
| \$STRNG | = 000004    | #8-177  |
| \$SUBXP | = 000010    | #8-177  |
| \$L INX | = ***** GX  | *7-88   |
| \$ENODC | = ***** GX  | *7-97   |
| \$FLG   | = 177777    | #8-177  |
| \$KEY   | = 177777    | #8-177  |
| .PNUMB  | = ***** GX  | 7-84      7-87      7-88      7-93      7-96      7-97      7-102      7-105      7-110<br>7-113      7-118      7-121      7-126      7-129      7-134      7-137      7-142      7-145<br>7-151      7-154      7-160      7-163      7-169 |
| .TPARS  | = ***** GX  | 6-68  |

B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
E  
C  
D  
E  
F  
C  
H  
I  
K  
L  
M  
N  
E  
C  
D  
E  
F  
C  
H  
I  
K  
L  
M  
N  
E  
C  
D  
E  
F  
C  
H  
I

CFGX3P

CF2AC1

CF2AC2

CF2AC3

J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
B  
C  
D

CF2AC4

CF2AC5

CF2AC6

E F G H I J K L M N B C D E F G H I J K L M N B C D E F G H I J K L

CF2AC7

CF2AC8

CF2AC9

|   |    |
|---|----|
| M | 14 |
| N | 14 |
| B | 15 |
| C | 15 |
| D | 15 |
| E | 15 |
| F | 15 |
| G | 15 |
| H | 15 |
| I | 15 |
| J | 15 |
| K | 15 |
| L | 15 |
| M | 15 |
| N | 15 |
| B | 16 |
| C | 16 |
| D | 16 |
| E | 16 |
| F | 16 |
| G | 16 |
| H | 16 |
| I | 16 |
| J | 16 |
| K | 16 |
| L | 16 |
| M | 16 |

CF2DEC

CF 2UMR

••FILE••ID••CFPCUG

B 1

```
CCCCCCCC  FFFFFFFF  PPPPPPP  CCCCCCCC  UU      UU      GGGGGGGG
CCCCCCCC  FFFFFFFF  PPPPPPP  CCCCCCCC  UU      UU      GGGGGGGG
CC         FF         PP         CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CC         FFFFFFFF  PPPPPPP  CC         UU      UU      GG
CC         FFFFFFFF  PPPPPPP  CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CC         FF         PP         CC         UU      UU      GG
CCCCCCCC  FF         PP         CCCCCCCC  UUUUUUUUUU  GGGGGG
CCCCCCCC  FF         PP         CCCCCCCC  UUUUUUUUUU  GGGGGG
```

....  
....  
....  
....

```
LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LL          SSSSSS    TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT
```

```

320 .SBTTL CHKUCUG - CHECK FOR UNIQUE CUG NAME
321
322 ;+
323
324 CHKUCUG - CHECK FOR UNIQUE CUG NAME
325
326 INPUTS:
327 CUGNAM - SPECIFIED CUG NAME
328
329 OUTPUTS:
330 CARRY CLEAR - SPECIFIED CUG NAME IS UNIQUE
331 CARRY SET - SPECIFIED CUG NAME IS NOT UNIQUE
332 RO DESTROYED
333
334
335 000646 017700 000000G CHKUCUG: MOV @PSIPT,RO ; GET ADDRESS OF PSI HOME BLOCK
336 000652 062700 000010 ADD #H$CUG,RO ; POINT TO CUG LISTHEAD
337
338 000656 SWSTK$ 30$ ;: ENTER SYSTEM STATE
339 000662 SAVMAP ;: SAVE CURRENT MAPPING
340 000666 011000 10$: MOV (R0),RO ;: GET ADDRESS OF NEXT BLOCK IN LIST
341 000670 001430 BEQ 25$ ;: BR IF END OF LIST (NAME IS UNIQUE)
342 000672 010046 MOV RO,-(SP) ;: SET UP FOR CONVERSION
343 000674 CALL $CEACK ;: CONVERT TO MAPPED ADDRESS
344 000700 012600 MOV (SP)+,RO ;: RETRIEVE MAPPED ADDRESS
345 000702 026067 000012 000000G CMP G$DTE(RO),DTEDES ;: SAME DTE AS NEW BLOCK?
346 000710 001366 BNE 10$ ;: NO, SO DON'T CHECK NAME
347 000712 012702 000000G MOV #CUNMMX,R2 ;: GET LENGTH OF CUG NAME
348 000716 012701 000000' MOV #CUCNAM,R1 ;: POINT TO CUGNAM
349 000722 010003 MOV RO,R3 ;: POINT TO CUG NAME BLOCK
350 000724 062703 000002 ADD #G$NAM,R3 ;: POINT TO CUG NAME
351 000730 122123 20$: CMPB (R1)+,(R3)+ ;: IS THE NAME UNIQUE?
352 000732 001355 BNE 10$ ;: BR IF YES
353 000734 005302 DEC R2 ;: MORE TO CHECK?
354 000736 003374 BGT 20$ ;: BR IF YES
355 000740 RETC RO ;: ELSE ERROR - SET USER C-BIT
356 000752 25$: RESMAP ;: RESTORE PREVIOUS MAPPING
357 000756 30$: RETURN ;: RETURN TO USER STATE
358 000001 .END

```

```

163                                     .SBTTL DSC$DF STATE TABLES
164
165                                     :
166                                     : DSC$DF
167                                     :
168 000060                               STATES$ DSCDF
169 000060                               TRANS$ %DSC$DF%,1,SYNERR
170
171 000060                               STATES$
172 000060                               TRANS$ !END,$EXIT
173 000060                               TRANS$ $STRNG,,DCMSK ; CALL MASK
174
175 000060                               STATES$
176 000060                               TRANS$ <','>
177
178 000060                               STATES$ CALVAL
179 000060                               TRANS$ $STRNG,END,DCVAL ; CALL VALUE
180
181 000060                               STATES$ END
182 000060                               TRANS$ $EOS,$EXIT
183 000060                               TRANS$ <','>,$EXIT
184
185 000060                               STATES$

```

```

55      .SBTTL  MACRO DEFINITIONS
56
57      ;
58      ; LIBRARY MACROS
59      ;
60      .MCALL  RETC,EMSG$R,DSTDF$,SAVRG,RESRG,PHBDF$,NTLERS$
61      .MCALL  ISTAT$,STATE$,TRANS$
62
63      DSTDF$      ; DESTINATION DESCRIPTOR BLOCK OFFSETS
64      PHBDF$      ; PSI HOME BLOCK OFFSETS
65
66      ;
67      ; LOCAL MACRO DEFINITIONS
68      ;
69      .MACRO  SAVMAP
70      MOV    @KSAR5,-(SP)    ; SAVE APR 5
71      .ENDM
72
73      .MACRO  RESMAP
74      MOV    (SP)+,@KSAR5    ; RESTORE APR5
75      .ENDM

```



```

Symbol table
ALLDST 001104R      DS.X29= ***** GX      HF$GWY= 000010      L$ASG= 000000      V$$CTR= 001000
A$$CHK= 000000      D$ACL 000027      HF$HOS= 000004      L$DRV= 000000      X$$DBT= 000000
A$$CPS= 000000      D$UGL 000023      HF$XDF= 000020      L$SPI= 000001      $ALPHA= 000022
A$$PRI= 000000      D$UGN 000022      H$CUG 000010      M$$CRB= 000124      $ANY = 000020
A$$TRP= 000000      D$DATL 000030      H$DST 000012      M$$CRX= 000000      $BLANK= 000006
BIAS = 000002      D$DSL 000022      H$DST 000014      M$$FCS= 000000      $CAT5 = ***** GX
CEPR = ***** GX  D$DST 000010      H$LEN 000104      M$$GME= 000000      $CEACK= ***** GX
CFGBF = ***** GX D$FLEN 000032      H$GLT 000044      M$$NET= 000000      $DIGIT= 000024
CFGSZ = ***** GX D$GLEN 000032      H$GNAM 000050      M$$OVR= 000000      $DNUMB= 000014
CLIN = ***** GX  D$GVBL 000032      H$GNML= 000020      M$$SOF= 000000      $EOS = 000012
CHKDSN 001012R      D$LNK 000000      H$GPT 000046      N$$ACC= 000001      $ERRM7 000204R
CUNMMX= ***** GX D$NAM 000004      H$HITS 000034      N$$BUF= 000001      $ERRM9 000236R
C$CKP= 000000      D$NOD 000010      H$LEN 000044      N$$LDV= 000001      $ERRN4 000000R
C$SORE= 000400      D$OBJ 000003      H$LBDA 000070      N$$MCP= 000001      $ERRN5 000032R
C$SRSH= 177564      D$OBJ 000003      H$LBDA 000070      N$$MLL= 000001      $ERRN6 000062R
DSACUG= ***** GX D$PAL 000026      H$LDTE 000002      N$$MOV= 000010      $ERRN7 000110R
DSARCT= ***** GX D$PRI 000002      H$LEN 000042      N$$NCT= 000001      $ERRIT= ***** GX
DSASHI= ***** GX D$PRL 000024      H$LEN 000042      N$$PEM= 000001      $ERRM 000132R
DSASLO= ***** GX D$SHI 000020      H$LOTS 000032      PSIPT = ***** GX      $EXIT = 000000
DSCMCT= ***** GX D$SLO 000016      H$NETW 000024      P$P45= 000000      $FAIL = 177777
DSCVCT= ***** GX D$USL 000025      H$NML = 000006      P$WRD= 000000      $HEADR= ***** GX
D$FLG = ***** GX D$VBL 000032      H$NPT 000022      Q$DPT= 000010      $LAMDA= 000000
DSNAM 000572R      D$BUG= 177514      H$PTB 000020      RTSPC = ***** GX      $LINKX= ***** GX
DSNAMX= 000006      D$ISK= 000000      H$PVC 000006      R$R0 = 000002      $NUMBER= 000002
DSNM 000024R      D$SL11= 000001      H$RDTE 000004      R$R1 = 000004      $QDST 000314RG
DSOBJ 000536R      D$YNC= 000000      H$RNV 000042      R$R2 = 000006      $RAD50= 000016
DSPRI 000530R      D$YNM= 000000      H$SVC 000036      R$R3 = 000010      $STRNG= 000004
DSTDF 000000R      END 000062R      H$STR 000016      R$R4 = 000012      $SUBXP= 000010
DSTEND 000664RG      E$XPR= 000000      H$XAVL 000100      R$R5 = 000014      $XALOC= ***** GX
DSTINS 001370R      FMT10 = ***** GX      H$XBLA 000074      R$RDER= 000000      $$$FLG= 177777
DSTKMX= 000006      FMT8 = ***** GX      H$X29C 000040      R$R11= 000001      $$$KEY= 000002
DSTKW 000000RG      FM.10 = 000000      I$RAR= 000000      R$RNU= 000000      $$$R = 000010
DSTNAM= ***** GX  FM.8 = 000000      I$RDN= 000000      R$R11M= 000000      $$$STA= 000000
DSTOBJ= ***** GX  F$SLVL= 000001      K$AR5 = ***** GX      SPACE = 000040      $$$TMP= 000013R
DSTPRI= ***** GX  G$STPP= 000000      K$CNT= 177546      SYNERR= ***** GX      .PNUMB= ***** GX
DSTSK 000606R      G$STSS= 000000      K$CSR= 177546      S$WRG= 000000      .PNUMH= ***** GX
DSTST 000000RG      G$STTK= 000000      K$SLDC= 000000      S$YSZ= 007600      .PSTCN= ***** GX
DSTTSK= ***** GX  G$WRD= 000000      K$TPS= 000074      T$KMG= 000000      .PSTPT= ***** GX
DSTVAR= ***** GX  HF$DLM= 000002      LD$LP = 000000      T$MIN= 000000      .TPARS= ***** GX
DS.X25= ***** GX

```

```

. ABS. 000104 000 (PW,I,GBL,ABS,OVR)
      001456 001 (RW,I,LCL,REL,CON)
$STATE 000072 002 (RW,D,LCL,REL,CON)
$KTAB 000006 003 (RW,D,LCL,REL,CON)
$KSTR 000017 004 (RW,D,LCL,REL,CON)
Errors detected: 0

```

### \*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 14023 words ( 55 Pages)
Size of core pool: 15496 words ( 59 Pages)
Operating system: RSX-11M/PLUS

```

Elapsed time: 00:00:33.30

B 6

```

164          .SBTTL  TPARS STATE TABLES
165          :
166          : TPARS STATE TABLES
167          :
168          : ISTAT$  DTEST,DTEKW
169          :
170          : DTE$DF
171          :
172          STATES$ DTEDF
173          TRANS$ %DTE$DF%,,,1,SYNERR
174
175          STATES$
176          TRANS$ $DIGIT,,DTEAST          ; START OF DTE ADDRESS
177
178          STATES$ DTEAD
179          TRANS$ $DIGIT,DTEAD,DTEDIG      ; DTE ADDRESS
180          TRANS$ <','>,,DTEAEN          ; END OF DTE ADDRESS
181
182          STATES$
183          TRANS$ !LINID                  ; LINE-ID
184
185          STATES$
186          TRANS$ <','>
187
188          STATES$
189          TRANS$ $NUMBR,,HSHTBL          ; HASH TABLE SIZE
190
191          STATES$
192          TRANS$ <','>
193
194          STATES$
195          TRANS$ $NUMBR,,CNTTIM          ; COUNTER TIMER VALUE
196
197          STATES$
198          TRANS$ !END,$EXIT,DTEEND
199          TRANS$ <','>
200
201          STATES$
202          TRANS$ "ON",DTENW,,DE.ON,DTEFLG      ; STATE
203          TRANS$ "OFF",DTENW,,DE.OFF,DTEFLG
204
205          STATES$ DTENW
206          TRANS$ !END,$EXIT,DTEEND
207          TRANS$ <','>
208
209          STATES$
210          TRANS$ $STRNG,,DTENET          ; NETWORK NAME PARAMETER
211
212          STATES$ END
213          TRANS$ $EOS,$EXIT
214          TRANS$ <','>,$EXIT

```

C 6

CFPDTE CREATED BY MACRO ON 29-JUN-85 AT 00:17 PAGE 2 B 7  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES             |
|---------|------------|------------------------|
| LF.TIM  | = 000010   | #5-67                  |
| LF.UNL  | = 020000   | #5-67                  |
| LF.X2P  | = 000000   | #5-67                  |
| LINCTL  | = ***** GX | 17-502                 |
| LINNAM  | = ***** GX | 17-489                 |
| LINUNT  | = ***** GX | 17-504                 |
| LLCTA   | = ***** GX | 15-387                 |
| LNKEND  | = ***** GX | 15-409                 |
| LN\$OFF | = 000001   | 15-395                 |
| LN\$ON  | = 000000   | 15-398                 |
| LN.CLO  | = 000000   | #5-67                  |
| LN.DUM  | = 000005   | #5-67                  |
| LN.LOA  | = 000004   | #5-67                  |
| LN.LOO  | = 000003   | #5-67                  |
| LN.OAU  | = 000003   | #5-67                  |
| LN.OFF  | = 000001   | #5-67                  |
| LN.ON   | = 000000   | #5-67                  |
| LN.OOP  | = 000004   | #5-67                  |
| LN.OPE  | = 000001   | #5-67                  |
| LN.REF  | = 000002   | #5-67                  |
| LN.SER  | = 000002   | #5-67                  |
| LN.STA  | = 000017   | #5-67                  |
| LN.SUB  | = 000360   | #5-67                  |
| LN.TRI  | = 000006   | #5-67                  |
| L\$CTB  | 000030     | *15-404                |
| L\$CTEN | 000032     | *15-405                |
| L\$CTIM | 000040     | *15-391                |
| L\$DTEA | 000020     | 15-369 16-453          |
| L\$DTL  | 000017     | *15-375                |
| L\$LEN  | 000122     | 15-350 15-401          |
| L\$LLCH | 000016     | *15-390                |
| L\$NETW | 000114     | 15-378                 |
| L\$NMST | 000003     | *15-395 *15-398 15-399 |
| L\$OMST | 000002     | *15-399                |
| L\$SLN  | 000004     | *15-385                |
| L\$TCLZ | 000044     | *15-394                |
| L.COST  | 000015     | 45-67                  |
| L.CTL   | 000012     | #5-67 17-502           |
| L.CVA   | 177776     | #5-67                  |
| L.DDM   | 000002     | #5-67 17-500           |
| L.DDS   | 000004     | #5-67                  |
| L.DLC   | 000003     | #5-67                  |
| L.DLM   | 000006     | #5-67                  |
| L.DLS   | 000010     | #5-67                  |
| L.FLG   | 000000     | #5-67 17-509           |
| L.KRBA  | 000016     | #5-67                  |
| L.LEN   | = 000022   | #5-67                  |
| L.MPF   | 000022     | #5-67                  |
| L.NMST  | 000020     | #5-67                  |
| L.NSTA  | 000014     | #5-67                  |
| L.OWNR  | 000021     | #5-67                  |
| L.UNT   | 000013     | #5-67 17-504           |

```

210 .SBTTL PSN$DF ACTION ROUTINES
211
212 ; NETWORK NAME
213
214 PSNNM: TST @PSIPT ; HAVE WE ALREADY FOUND A PSN$DF?
215 000052 005777 000000G BNE 101$ ; BR IF YES - ERROR
216 000056 001032 MOV #PSTCN,R0 ; GET NUMBER OF CHARACTERS IN NAME
217 000060 016700 000000G MOV #FNAMMX,R2 ; GET MAXIMUM ALLOWABLE CHARACTERS
218 000064 012702 000006 CMP R0,R2 ; IS THIS A VALID COUNT
219 000070 020002 BHI 102$ ; BR IF NO
220 000072 101027 MOV #PSNNAM,R1 ; POINT TO TEMPORARY LOCATION FOR NAME
221 000074 012701 000000' SAVRG <R1> ; SAVE ADDRESS OF TEMPORARY LOCATION
222 000100 10$: MOV #SPACE,(R1)+ ; INITIALIZE NAME TO BLANKS
223 000102 112721 000040 DEC R2 ; MORE TO INITIALIZE?
224 000106 005302 BGT 10$ ; BR IF YES
225 000110 003374 RESRG <R1> ; RETRIEVE ADDRESS OF TEMPORARY NAME
226 000112 20$: MOV #PSTPT,R2 ; GET START OF SPECIFIED NAME
227 000114 016702 000000G MOVB (R2)+,(R1)+ ; STORE SPECIFIED NAME
228 000120 112221 DEC R0 ; MORE TO STORE?
229 000122 005300 BGT 20$ ; BR IF YES
230 000124 003375 CLR PSNFLG ; INITIALIZE FLAGS WORD
231 000126 005067 000006' CLR PSNLO ; CLEAR LOW SUBADDRESS VALUE
232 000132 005067 000010' CLR PSNHI ; CLEAR HIGH SUBADDRESS VALUE
233 000136 005067 000012' RETURN
234 000142
235
236 ; ERROR MESSAGES
237
238 101$: MSG$R P1 ; ERROR - ILLEGAL OCCURENCE OF PSN$DF
239 000152 102$: MSG$R P2 ; ERROR - ILLEGAL NETWORK NAME
240
241 ; NUMBER OF PORTS
242
243 PORT: TST #PNUMH ; IS THIS A SINGLE WORD VALUE?
244 000160 005767 000000G BNE 101$ ; BR IF NO - ERROR
245 000164 001011 MOV #PNUMB,R0 ; GET PORT NUMBER
246 000166 016700 000000G BLE 101$ ; BR IF ILLEGAL PORT NUMBER
247 000172 003406 CMP R0,#PORTMX ; IS THIS A LEGAL PORT NUMBER?
248 000174 020027 000400 BHI 101$ ; BR IF NO - ERROR
249 000200 101003 MOV R0,PORTNO ; SAVE PORT NUMBER
250 000202 010067 000000G RETURN
251 000206
252
253 ; ERROR MESSAGES
254
255 101$: MSG$R P3 ; ERROR - ILLEGAL NUMBER OF PORTS
256
257 ; FLAGS WORD
258
259 PSNFG: TST #PNUMH ; IS THIS A DOUBLE WORD VALUE?
260 000216 005767 000000G BNE 101$ ; BR IF YES - ERROR
261 000222 001004 MOV #PNUMB,PSNFLG ; SAVE FLAGS WORD
262 000224 016767 000000G 000006' RETURN
263 000232
264
265 ; ERRORS
266

```

```

97          .SBTTL LOCAL SYMBOL DEFINITIONS
98
99          ; LOCAL SYMBOL DEFINITIONS
100         ;
101         ;
102         000040      SPACE = 40          ; ASCII SPACE
103         000006      PVNAMX = 6         ; MAXIMUM LENGTH OF CIRCUIT ID
104
105         ;
106         ; SAVED REGISTER OFFSETS ON STACK FOR SWSTK$
107         ;
108         000002      R$R0 = 2           ; SAVED R0
109         000004      R$R1 = 4           ; SAVED R1
110         000006      R$R2 = 6           ; SAVED R2
111         000010      R$R3 = 10          ; SAVED R3
112         000012      R$R4 = 12          ; SAVED R4
113         000014      R$R5 = 14          ; SAVED R5

```

```

599                                     .SBTTL FNDHSH - FIND HASH TABLE ENTRY
600
601                                     ;+
602                                     : FNDHSH - FIND HASH TABLE ENTRY
603                                     :
604                                     : INPUTS:
605                                     :   PVCLCN - LOGICAL CHANNEL NUMBER
606                                     :   PVCPRT - PORT NUMBER
607                                     :
608                                     : OUTPUTS:
609                                     :   CARRY CLEAR - TABLE ENTRY AVAILABLE AND SET UP WITH LCN/PORT
610                                     :   CARRY SET - NO ENTRIES AVAILABLE
611                                     :   R0,R1,R2 DESTROYED
612                                     :   MAPPING PRESERVED
613                                     :
614                                     : -
615 FNDHSH: SAVMAP                      ;; SAVE MAPPING
616      MOV DTEDES,-(SP)                ;; GET UNMAPPED DTE DESCRIPTOR
617      CALL $CEACX                     ;; MAP TO LOCAL DTE (AND ALSO HASH TABLE)
618      TST (SP)+                       ;; CLEAN STACK
619      CLR R2                          ;; INITIAL OFFSET FOR HASH TABLE INSERTION
620      MOV PVCLCN,R0                   ;; GET LOGICAL CHANNEL NUMBER
621      MOV HSHSZ,R1                   ;; GET NUMBER OF ENTRIES IN HASH TABLE
622      NEG R1                          ;; GET MASK FOR HIGH BITS
623      BIC R1,R0                      ;; COMPUTE ENTRY NUMBER (MODULO TABLE SIZE)
624      MOV R0,R1                      ;; SAVE ENTRY NUMBER
625      ASL R0                          ;; MULTIPLY BY 4 TO DETERMINE OFFSET
626      ASL R0                          ;; ... INTO HASH TABLE
627      ADD HSHADD,R0                  ;; DETERMINE ENTRY ADDRESS IN HASH TABLE
628      TST 2(R0)                      ;; IS THIS ENTRY AVAILABLE?
629      BEQ 20$                       ;; BR IF YES (CLEARS CARRY)
630      INC R2                          ;; ELSE, COLLISION - UPDATE OFFSET
631      CMP HSHSZ,R2                  ;; ANY ENTRIES AVAILABLE?
632      BLO 30$                       ;; BR IF NO (SETS CARRY)
633      ADD R2,R1                      ;; CALCULATE ENTRY NUMBER + OFFSET
634      MOV R1,R0                      ;; SAVE ENTRY NUMBER + OFFSET
635      BR 10$                        ;; FIND NEXT AVAILABLE ENTRY
636      MOV PVCLCN,(R0)+               ;; STORE LOGICAL CHANNEL NUMBER IN HASH TABLE
637      MOVB PVCPRT,(R0)               ;; STORE PORT NUMBER IN LOW BYTE OF HASH TABLE
638      RESMAP                         ;; RESTORE MAPPING
639      RETURN

```

B 11

77  
78  
79  
80  
81  
82  
83

.SBTTL LOCAL SYMBOL DEFINITIONS

;  
; LOCAL SYMBOL DEFINITIONS  
;

000006  
000040

RDIEMX = 6  
SPACE = 40

; MAXIMUM LENGTH OF REMOTE DTE NAME  
; ASCII SPACE CHARACTER

C 11

CFPX29 - CONFIG FILE ACTION ROU MACRO V05.03b Saturday 29-Jun-85 00:20 <sup>B 12</sup>  
Table of contents

|    |     |                                       |
|----|-----|---------------------------------------|
| 5- | 52  | MACRO DEFINITIONS                     |
| 6- | 63  | LOCAL DATA STORAGE AND ERROR MESSAGES |
| 7- | 89  | LOOK FOR X29\$DF MACRO                |
| 8- | 119 | TPARS STATE TABLES                    |
| 9- | 150 | X29\$DF ACTION ROUTINES               |



.TITLE CFPX3P - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 31-MAR-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSx V1.0

MACPO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |         |         |         |         |         |        |        |        |         |
|------------|------------|---------|---------|---------|---------|---------|--------|--------|--------|---------|
| CALL       | 8-130      | 11-208  | 11-220  | 11-235  | 11-247  | 12-287  | 12-288 | 12-296 |        |         |
| CHNDF\$    | #5-60      | 5-62    |         |         |         |         |        |        |        |         |
| DBGTP\$    | #9-147     | #10-189 | #10-199 |         |         |         |        |        |        |         |
| DTEDF\$    | #5-60      | 5-63    |         |         |         |         |        |        |        |         |
| EMSG&R     | #5-60      | 8-139   | 11-215  | 11-229  | 11-230  | 11-242  | 11-256 | 11-257 | 12-321 | 12-322  |
| ISTAT\$    | #5-60      | 9-147   |         |         |         |         |        |        |        |         |
| MTRAN\$    | #9-147     |         |         |         |         |         |        |        |        |         |
| NTLER\$    | #5-60      | 7-99    |         |         |         |         |        |        |        |         |
| RESMAP     | #5-72      | 12-313  |         |         |         |         |        |        |        |         |
| RETC       | #5-60      | 12-290  |         |         |         |         |        |        |        |         |
| RETURN     | 8-134      | 11-211  | 11-225  | 11-231  | 11-252  | 12-273  | 12-314 | 12-317 |        |         |
| SAVMAP     | #5-68      | 12-294  |         |         |         |         |        |        |        |         |
| STAT\$     | #5-60      | 9-152   | #9-155  | #9-158  | #9-161  | #9-164  | #9-167 | #9-170 | #9-173 | 10-181  |
|            | #10-184    | #10-187 | #10-191 | #10-197 | #10-201 |         |        |        |        |         |
| SWSTK\$    | 12-287     |         |         |         |         |         |        |        |        |         |
| TRANS      | #5-60      | #9-153  | #9-156  | #9-159  | #9-162  | #9-165  | #9-168 | #9-171 | #9-174 | #10-182 |
|            | #10-185    | #10-188 | #10-189 | #10-192 | #10-198 | #10-199 |        |        |        |         |

ELTIME - CALCULATE ELAPSED TIME MACRO V05.03b Saturday 29-Jun-85 <sup>B.15</sup> 17:42  
Table of contents

5- 57 ELTIME - CALCULATE ELAPSED TIME

```

151                                     .SBTTL $READX - READ INTO EXTENDED MEMORY
152                                     :+
153                                     $READX - READ A LOGICAL BLOCK INTO EXTENDED MEMORY
154                                     :
155                                     INPUTS:
156                                     R0=BUFFER BIAS
157                                     :
158                                     OUTPUTS:
159                                     C-BIT=SUCCESS/FAILURE
160                                     R0=DESTROYED
161                                     :
162                                     NOTE: (RSX11M ONLY)
163                                     THIS ROUTINE USES APRI TO MAP THE BUFFER AREA. INTERRUPTS MUST BE
164                                     DISABLED WHILE THE QIO IS ISSUED TO PREVENT A TASK SWITCH WHICH WOULD
165                                     RESTORE THE ORIGINAL (INCORRECT) CONTENTS OF APRI.
166                                     :
167                                     $READX::
168                                     :
169                                     .IF DF R$$$11M & M$$$MGE
170                                     :
171                                     SWSTK$ 10$ ; ENTER SYSTEM STATE
172                                     BIS #PR7,20(SP) ; RETURN TO USER STATE WITH INTERRUPTS DISABLD
173                                     MOV USAR0,R1 ; GET UISAR0 VALUE
174                                     ADD #2,R1 ; GET USER APRI ADDRESS
175                                     MOV (R1),UISAV ; SAVE CURRENT APRI
176                                     MOV R0,(R1) ; SET UP TRANSFER BIAS
177                                     RETURN
178                                     :
179                                     10$: MOV #20000,R0 ;;; SET NEW BUFFER ADDRESS
180                                     CALL $READ ;;;
181                                     ADC R0 ;;; SAVE C-BIT
182                                     :
183                                     SWSTK$ 20$ ; ENTER SYSTEM STATE
184                                     MOV USAR0,R1 ; GET USER APRI ADDRESS
185                                     ADD #2,R1 ; GET USER APRI ADDRESS
186                                     MOV UISAV,(R1) ; RESTORE OLD APRI CONTENTS
187                                     BIS R0,20(SP) ; RESTORE USER MODE PSW ON NEXT RETURN
188                                     :
189                                     20$: RETURN
190                                     :
191                                     .ENDC
192                                     :
193                                     .IF DF R$$$11D ! I$$$AS
194                                     :
195                                     MOV #77406,-(SP) ; SET NEW PDR VALUE - 4K READ/WRITE
196                                     MOV R0,-(SP) ; SET NEW PAR VALUE
197                                     CALL ..SPD3 ; CHANGE MAPPING IN HEADER
198                                     MOV #60000,R0 ; SET NEW BUFFER ADDRESS
199                                     CALL $READ
200                                     ROR 4(SP) ; SAVE C-BIT
201                                     CALL ..SPD3 ; RESTORE MAPPING
202                                     CMP (SP)+,(SP)+ ; PURGE STACK
203                                     ASL (SP) ; RESTORE C-BIT
204                                     RETURN
205                                     :
206                                     .ENDC

```

CFPCUG - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:15 C 1  
Table of contents

|     |     |                                    |
|-----|-----|------------------------------------|
| 5-  | 52  | MACRO DEFINITIONS                  |
| 6-  | 74  | LOCAL SYMBOL DEFINITIONS           |
| 7-  | 81  | LOCAL DATA                         |
| 8-  | 97  | ERROR MESSAGES                     |
| 9-  | 111 | LOOK FOR CUG\$DF MACRO             |
| 10- | 140 | CUG\$DF STATE TABLES               |
| 11- | 177 | CUG\$DF ACTION ROUTINES            |
| 14- | 320 | CHKCUG - CHECK FOR UNIQUE CUG NAME |

```

Symbol table
A$$$CHK= 000000 D$$$YNC= 000000 H$GPT 000046 M$$$CRX= 000000 $ALPHA= 000022
A$$$CPS= 000000 D$$$YNM= 000000 H$HITS 000034 M$$$FCS= 000000 $ANY = 000020
A$$$PRI= 000000 END 000050R 003 H$HLEN 000044 M$$$MGE= 000000 $BLANK= 000006
A$$$TRP= 000000 E$$$XPR= 000000 H$LBDA 000070 M$$$NET= 000000 $CEACX= ***** GX
CERR = ***** GX FMT10 = ***** GX H$LBDA 000072 M$$$QVR= 000000 $DIGIT= 000024
CFGBF = ***** GX FMT8 = ***** GX H$LDTE 000002 M$$$NEXT = ***** GX $DNUMB= 000014
CFGSZ = ***** GX FM.10 = 000000 H$LEN 000042 M$$$ACC= 000001 $EOS = 000012
CFLIN = ***** GX FM.8 = 000000 H$LOTS 000032 M$$$BUF= 000001 $ERRM1 000110R 002
CHKCUG 000646R F$$$LVL= 000001 H$NETW 000024 M$$$LDV= 000001 $ERRN2 000016R 002
COUNT = ***** GX GF$BUG= 000001 H$NML = 000006 M$$$MCP= 000001 $ERRN3 000040R 002
CUEND 000406R G$CUG 000010 H$NPT 000022 M$$$MLL= 000001 $ERRP4 000064R 002
CUG 000364R G$DTE 000012 H$PTB 000020 M$$$MOV= 000010 $ERRIT= ***** GX
CUGDF 000000R 003 G$FLG 000014 H$PVC 000006 M$$$NCT= 000001 $EXIT = 000000
CUGDIG 000174R G$GNAM 000016 H$RDTE 000004 M$$$PEM= 000001 $FAIL = 177777
CUGEND= 000011R 002 G$LEN 000016 H$RNW 000042 PCKBCD= ***** GX $HEADR= ***** GX
CUGFLG 000014R 002 G$LNK 000000 H$SVC 000036 PSIPT = ***** GX $LAMD= 000000
CUGKW 000000RG 004 G$NAM 000002 H$TRB 000016 P$P45= 000000 $NUMBR= 000002
CUGNAM 000000R 002 G$NML 000015 H$XAVL 000100 P$WRD= 000000 $QUG 000000RG
CUGNLN= 000004 G$$$TPP= 000000 H$XBIA 000074 Q$OPT= 000010 $RAD50= 000016
CUGNUM 000006R 002 G$$$TSS= 000000 H$X29C 000040 RTSPC = ***** GX $SAVRG= ***** GX
CUGPCK 000012R 002 G$$$TTK= 000000 I$$$RAR= 000000 R$DER= 000000 $STRNG= 000004
CUGST 000000RG 003 G$$$WRD= 000000 I$$$RDN= 000000 R$K11= 000001 $SUBXP= 000010
CUNAM 000052R HF$DLM= 000002 K$AR5 = ***** GX R$SND= 000000 $XALOC= ***** GX
CUNBEN 000220R HF$GWY= 000010 K$CNT= 177546 R$11M= 000000 $$$$FLG= 177777
CUNEST 000136R HF$HOS= 000004 K$CLDL 000000 SPACE = 000040 $$$$KEY= 000000
CUNMMX= ***** GX HF$XDF= 000020 K$TPS= 000074 STRNXT= ***** GX $$$R = 000010
CUNXT 000020R 003 H$CUG 000010 LD$LP = 000000 SYNERR= ***** GX $$$STA= 000000
C$$$CKP= 000000 H$DST 000012 LNKEND= ***** GX $$$WRG= 000000 005
C$$$ORE= 000400 H$D29 000014 L$ASG= 000000 $$$YSZ= 007600
C$$$RSH= 177564 H$FLG 000000 L$DRV= 000000 T$$$KMG= 000000
C$$$DES= ***** GX H$GLEN 000104 L$P11= 000001 T$$$MIN= 000000
D$$$BUG= 177514 H$GLT 000044 L$$$11R= 000000 V$$$CTR= 001000
D$$$ISK= 000000 H$GNAM 000050 M$$$CRB= 000124 X$$$DBT= 000000
D$$$L11= 000001 H$GNML= 000020

```

```

. ABS. 000104 000 (RW,I,GBL,ABS,OVR)
      000760 001 (RW,I,LCL,REL,CON)
DATA 000162 002 (RW,D,LCL,REL,CON)
$STATE 000060 003 (RW,D,LCL,REL,CON)
$KTAB 000002 004 (RW,D,LCL,RFL,CON)
$KSTR 000007 005 (RW,D,LCL,REL,CON)
Errors detected: 0

```

# \*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 13410 Words ( 53 Pages)
Size of core pool: 14440 Words ( 55 Pages)
Operating system: RSX-11M/PLUS

```

Elapsed time: 00:00:27.02  
SY:CFPCUG.V2,[132,134]CFPCUG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETL.B/ML,[130,10]RSXMCM/PA:1,[132,10]CFPCUG

```

187
188 .SBTTL DSA$DF ACTION ROUTINES
189
190 ;
191 ; START OF REMOTE DTE ADDRESS
192 ;
193 000060 012700 000000G DARDST: MOV #DARMLN,R0 ; GET LENGTH OF REMOTE DTE ADDRESS
194 000064 012701 000000G MOV #DSARDT,R1 ; GET ADDRESS OF START OF BUFFER
195 000070 105021 10$: CLRB (R1)+ ; INITIALIZE BUFFER
196 000072 005300 DEC R0 ; MORE TO INITIALIZE?
197 000074 003375 BGT !0$ ; BR IF YES
198 000076 105067 000000G CLRB COUNT ; INITIALIZE DIGIT COUNT
199 000102 012767 000000G 000000G MOV #DSARDT,NEXT ; START AT BEGINNING OF BUFFER
200 000110 CALL STRNXT ; STORE CHARACTER
201 000114 RETURN
202
203 ;
204 ; PROCESS REMOTE DTE DIGIT
205 ;
206 000116 026727 000000G 000000G DARDNO: CMP NEXT,#DARDEN ; IS ADDRESS TOO LONG?
207 000124 101003 BHI !01$ ; BR IF YES
208 000126 CALL STRNXT ; STORE CHARACTER
209 000132 RETURN
210
211 ;
212 ; ERRORS
213 ;
214 000134 101$: MSG$R N8 ; ILLEGAL REMOTE DTE ADDRESS
215
216 ;
217 ; END OF REMOTE DTE ADDRESS
218 ;
219 000142 004567 000000G DARDND: JSR R5,$SAVRG ; SAVE R3-R5
220 000146 012700 000000G MOV #<DARMLN/2>+1,R0 ; GET LENGTH OF PACKED ADDRESS
221 000152 012704 000000G MOV #DSARDT,R4 ; POINT TO NUMBER TO CONVERT
222 000156 012705 000000G MOV #DSARDP,R5 ; POINT TO STORAGE FOR PACKED NUMBER
223 000162 CALL PCKBCD ; PACK REMOTE DTE ADDRESS IN BCD
224 000166 116767 000000G 000000G MOVB COUNT,DSARCT ; STORE NUMBER OF DIGITS IN DTE ADDRESS
225 000174 RETURN

```

```

76                                     .SBTTL  LOCAL SYMBOL DEFINITIONS
77
78                                     ;
79                                     ; LOCAL SYMBOL DEFINITIONS
80                                     ;
81                                     ;
82      - 000040      SPACE = 40          ; ASCII SPACE
83      000006      DSNAMX = 6          ; MAXIMUM LENGTH OF DESTINATION NAME
84      000006      DSTKMX = 6          ; MAXIMUM LENGTH OF OBJECT TASK NAME
85
86                                     ;
87                                     ; SAVED REGISTER OFFSETS ON STACK FOR SWSTK$
88                                     ;
89      000002      R$R0 = 2            ; SAVED R0
90      000004      R$R1 = 4            ; SAVED R1
91      000006      R$R2 = 6            ; SAVED R2
92      000010      R$R3 = 10           ; SAVED R3
93      000012      R$R4 = 12           ; SAVED R4
94      000014      R$R5 = 14           ; SAVED R5
95
96      000000      BIAS = 0            ;
97      000002      .IF DF M$MGE      ; SAVED KISAR6 MAPPING
98                                     .ENDC

```



CFPDST - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:16 Page 16-2  
Symbol table

SY:CFPDST.V2,[132,134]CFPDST/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFPDST

```
LINE-ID SUBEXPRESSION
216 .SBITL LINE-ID SUBEXPRESSION
217
218 ;
219 ; LINE-ID SUBEXPRESSION
220 ;
221 000052 STATES$ LINID
222 000052 TRANS$ $RAD50,,DEVNAM ; DEVICE NAME
223
224 000052 STATES$
225 000052 TRANS$ <'->
226
227 000052 STATES$
228 000052 TRANS$ $NUMBR,,DEVCTL ; CONTROLLER NUMBER
229
230 000052 STATES$
231 000052 TRANS$ <'->,LINID1
232 000052 TRANS$ $LAMDA,$EXIT
233
234 000052 STATES$ LINID1
235 000052 TRANS$ $NUMBR,$EXIT,DEVUNT ; UNIT NUMBER
236
237 000052 STATES$
```

SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL   | VALUE      | REFERENCES                          |
|----------|------------|-------------------------------------|
| NEXT     | = ***** GX | *12-250 12-257                      |
| NUMDTE   | = 000020   | #6-87 15-312                        |
| NS\$VCT  | = *****    | 15-355 16-459 17-512                |
| PCKBCD   | = ***** GX | 12-273                              |
| PSIPT    | = ***** GX | 12-275 15-406 16-439                |
| RTSPC    | = ***** GX | 8-124 8-125 8-126 8-127 8-128 8-129 |
| R\$RO    | = 000002   | #6-92                               |
| R\$R1    | = 000004   | #6-93                               |
| R\$R2    | = 000006   | #6-94                               |
| R\$R3    | = 000010   | #6-95                               |
| R\$R4    | = 000012   | #6-96 *16-461 *17-513               |
| R\$R5    | = 000014   | #6-97                               |
| SF.ACT   | = 000200   | #5-67                               |
| SF.ENA   | = 000100   | #5-67                               |
| SF.LPB   | = 000004   | #5-67                               |
| SF.MFL   | = 000040   | #5-67                               |
| SF.PAC   | = 000020   | #5-67                               |
| SF.REA   | = 000010   | #5-67                               |
| SF.SER   | = 000001   | #5-67                               |
| SF.SVC   | = 000002   | #5-67                               |
| SF.UNL   | = 000040   | #5-67                               |
| SLTMA    | = ***** GX | 17-497                              |
| SLTNM    | = ***** GX | 17-496                              |
| SPACE    | = 000040   | #6-84                               |
| STRNXT   | = ***** GX | 12-251 12-259                       |
| SYNERR   | = ***** GX | *9-152 *9-155                       |
| \$S\$BAS | = *****    | 8-124 8-125 8-126 8-127 8-128       |
| S.COST   | 000001     | #5-67                               |
| S.FLG    | 000000     | #5-67                               |
| S.LEN    | 000004     | #5-67                               |
| S.NMST   | 000002     | #5-67                               |
| S.OWNR   | 000003     | #5-67                               |
| ZTIME    | = ***** GX | 15-392                              |
| \$ALPHA  | = 000022   | #10-168                             |
| \$ANY    | = 000020   | #10-168                             |
| \$BLANK  | = 000006   | #10-168                             |
| \$CEACX  | = ***** GX | 15-361 16-449                       |
| \$DIGIT  | = 000024   | #10-168                             |
| \$DNUMB  | = 000014   | #10-168                             |
| \$EOS    | = 000012   | #10-168                             |
| \$ERRO0  | 000154 R   | #8-127 14-334                       |
| \$ERRO1  | 000214 R   | #8-128 15-420                       |
| \$ERRO2  | 000266 R   | #8-129 15-421                       |
| \$ERRO3  | 000036 R   | #8-124 12-264 15-418                |
| \$ERRO4  | 000070 R   | #8-125 13-302                       |
| \$ERRO5  | = ***** GX | 13-314                              |
| \$ERRO7  | 000120 R   | #8-126 15-419                       |
| \$ERRIT  | = ***** GX | 9-162                               |
| \$EXIT   | = 000000   | #10-168                             |
| \$FAIL   | = 177777   | #10-168                             |
| \$GPM    | = *****    | 10-168                              |

267 000234

101\$: EMSG\$R P4

; ERROR - ILLEGAL FLAGS WORD

```

115          .SBTTL  LOCAL DATA
116
117          ;
118          ; LOCAL DATA
119          ;
120
121          .PSECT  DATA,D
122
123          .NLST  BEX
124
125          ;
126          ; LOCAL DATA FOR PVC$DF MACRO
127          ;
128          000000      PRTADD: .BLKW  1          ; ADDRESS OF ENTRY IN PORT TABLE
129          000002      PVCLCN: .BLKW  1          ; LOGICAL CHANNEL NUMBER
130          000004      PVCCTM: .BLKW  1          ; COUNTER TIMER
131          000006      PVCPSZ: .BLKW  1          ; MAXIMUM BLOCK SIZE
132          000010      PVCWSZ: .BLKW  1          ; WINDOW SIZE
133          000012      PVCOWN: .BLKW  1          ; OWNER PROCESS
134          000014      DLMPOV: .BLKW  1          ; ADDRESS OF DLM PDV
135          000016      DLMSLT: .BLKW  1          ; ADDRESS OF DLM SLT
136          000020      FLAG:   .BLKW  1          ; LOCAL FLAGS WORD
137          000001      PF.DLM = 1          ; PVC IS A DLM CIRCUIT
138          000022      PVCPRT: .BLKB  1          ; PORT NUMBER OF FREE ENTRY
139          000023      PVCFLG: .BLKB  1          ; PVC FLAGS BYTE
140          000024      PVCNAM: .BLKB  6          ; CIRCUIT IDENTIFICATION
141          000032      DLMCTL: .BLKB  1          ; DLM'S CONTROLLER NUMBER
142          000033      DLMUNT: .BYTE  0          ; DLM'S UNIT NUMBER (ALWAYS 0, MUST BE MTP)
143          000034      DLMSTA: .BLKB  1          ; DLM'S STATION NUMBER
144          000035      DLMIND: .BLKB  1          ; DLM'S PDV INDEX
145
146          ; STATE VALUE FOR "DATA TRANSFER STATE" FOR XCB
147          ; GLOBAL PATCHABLE JUST IN CASE ...
148
149          000036      000003      SP$P4:: .WORD  3          ; VALUE FOR DATA TRANSFER STATE
150

```

```

641                                     .SBTTL  BIASX - BIAS XPOOL ADDRESS FOR APR6
642
643                                     ;+
644                                     ; BIASX - BIAS XPOOL ADDRESS FOR APR6
645                                     ;
646                                     ; INPUTS:
647                                     ;   R0 = ADDRESS (POOL OR APR5)
648                                     ;
649                                     ; OUTPUTS:
650                                     ;   R0 = ADDRESS (POOL OR APR6)
651                                     ;
652                                     ; -
653
654                                     BIASX:
655                                     CMP     R0,#120000      ; POOL ADDRESS?
656                                     BLO     10$             ; YES, LEAVE AS IS
657                                     BIC     #160000,R0       ; CLEAR APR BITS
658                                     BIS     #140000,R0       ; BIAS FOR APR6
659                                     10$:  RETURN           ; RETURN

```

```

LOCAL DATA
85 .SBTTL LOCAL DATA
86
87 .NLIST BEX
88 .PSECT DATA,D
89
90 ; LOCAL DATA FOR RDT$DF MACRO
91
92 RDTNAM: .BLKB 6 ; REMOTE DTE NAME
93 RDTLN = 15.
94 RDTE: .BLKB RDTLN ; REMOTE DTE ADDRESS
95 RDTE: .BLKB RDTLN ; END OF DTE ADDRESS
96 RDTEP: .BLKB <RDTLN+1>/2 ; REMOTE DTE ADDRESS IN BCD
97 .EVEN
000000
000017
000006
000024

```

.TITLE CFPX29 - CONFIG FILE ACTION ROUTINES  
.IDENT /V05.00/

..COPYRIGHT (C) 1981, 1982, 1983, 1985, BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

..THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

..THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

..DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

..MODULE DESCRIPTION:

..NTL - CONFIG FILE ACTION ROUTINES FOR X.29 SUPPORT

..DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING GUEST STAR

..IDENT HISTORY:

- 1.00 10-JUL-81  
DECNET-11M V3.1  
MODULE CREATED
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0



```
55 .SBTTL MACRO DEFINITIONS
56
57 ;
58 ; LIBRARY MACROS
59 ;
60 .MCALL RETC,EMSG$R,CHNDF$,DTEDF$,NTLER$,TRANS$,STAT$,ISTAT$
61
62 000000 CHNDF$ ; DEFINE CHANNEL TABLE OFFSETS
63 000000 DTEDF$ ; DEFINE DTE DESCRIPTOR OFFSETS
64
65 ;
66 ; LOCAL MACRO DEFINITIONS
67 ;
68 .MACRO SAVMAP
69 MOV @KSAR5,-(SP) ; SAVE APR 5
70 .ENDM
71
72 .MACRO RESMAP
73 MOV (SP)+,@KSAR5 ; RESTORE APR5
74 .ENDM
75
```

\*\*FILE\*\*ID\*\*CTBIO

```

CCCCCCCC  TTTTTTTTTT  BBBB BBBB  IIIIII  000000
CCCCCCCC  TTTTTTTTTT  BBBB BBBB  IIIIII  000030
CC          TT      BB      BB      II      00      00
CC          TT      BB      BB      II      00      00
CC          TT      BB      BB      II      00      00
CC          TT      BB      BB      II      00      00
CC          TT      BBBB BBBB  II      00      00
CC          TT      BBBB BBBB  II      00      00
CC          TT      BB      BB      II      00      00
CC          TT      BB      BB      II      00      00
CC          TT      BB      BB      II      00      00
CC          TT      BB      BB      II      00      00
CC          TT      BB      BB      II      00      00
CCCCCCCC  TT      BBBB BBBB  IIIIII  000000
CCCCCCCC  TT      BBBB BBBB  IIIIII  000000

```

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LL          SSSSSS    TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT
LLLLLLLLLL  SSSSSSSS  TT

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55

```
.TITLE ELTIME - CALCULATE ELAPSED TIME
.IDENT /V05.00/

: COPYRIGHT (C) 1979, 1980, 1982, 1983, 1984, 1985 BY
: DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
:
: THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
: ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
: INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
: COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
: OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
: TRANSFERRED.
:
: THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
: AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
: CORPORATION.
:
: DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
: SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
:
: MODULE DESCRIPTION:
:
:   NETWORK MANAGEMENT - CALCULATE ELAPSED TIME
:
: DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
:
: IDENT HISTORY:
:
: 1.00 14-DEC-79
:       DECNET-11M/S V3.0
:       DECNET-11M-PLUS V1.0
:
: 2.00 16-APR-82
:       DECNET-11M V3.1
:       DECNET-11M-PLUS V1.1
:
: 4.00 07-NOV-83
:       DECNET-11M V4.0
:       DECNET-11M-PLUS V2.0
:
: 4.01 17-OCT-84          RJK
:       Change Symbol: for Vectoring Comm Exec
:
: 5.00 22-JUL-85
:       DECnet-11M/S V4.2
:       DECnet-11M-Plus V3.0
:       DECnet-Micro/RSX V1.0
```

```

208                                     .SBTTL $READ - READ LOGICAL BLOCK
209                                     ;+
210                                     ; $READ - READ A LOGICAL BLOCK
211                                     ;
212                                     ; INPUTS:
213                                     ; RO=BUFFER ADDRESS
214                                     ; $LLEN=BUFFER LENGTH
215                                     ; $LBN=LOGICAL BLOCK NUMBER
216                                     ;
217                                     ; OUTPUTS:
218                                     ; C-BIT=SUCCESS/FAILURE
219                                     ;
220 000152                                     $READ::
221 000152 010067 000030'                     MOV     RO,$LBUF          ;;; SET BUFFER ADDRESS
222 000156                                     DIR$    #LBNDPB          ;;; READ LOGICAL BLOCK
223 000164 103404                                     BCS     10$              ;;; IF CS, DPB ERROR
224 000166 105767 '00044'                       TSTB    $IOSB            ;;; WAS I/O SUCCESSFUL? (CLEARS C-BIT)
225 000172 100001                                     BPL     10$              ;;; IF PL, YES
226 000174 000261                                     SEC                                     ;;; INDICATE FAILURE
227 000176                                     10$:    RETURN
228
229                                     .END
230 000001

```

```

      .TITLE  CFPCUG - INITIAL CONFIG FILE SCAN ACTION ROUTINES
      .IDENT  /V05.00/

:  COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
:  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

:  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
:  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
:  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
:  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
:  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
:  TRANSFERRED.

:  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
:  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
:  CORPORATION.

:  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
:  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

:  MODULE DESCRIPTION:

:  NTL - CONFIG FILE ACTION ROUTINES

:  DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

:  IDENT HISTORY:

:  1.00  31-MAR-81
:         DECNET-11M/S V3.0
:         DECNET-11M-PLUS V1.0

:  3.00  16-APR-82
:         DECNET-11M V3.1
:         DECNET-11M-PLUS V1.1

:  4.00  07-NOV-83
:         DECNET-11M V4.0
:         DECNET-11M-PLUS V2.0

:  5.00  22-JUL-85
:         DECnet-11M/S V4.2
:         DECnet-11M-Plus V3.0
:         DECnet-Micro/RX V1.0

```

CFPCUG      CREATED BY MACRO ON 29-JUN-85 AT 00:15      PAGE 1      D 2

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|---|
| CERR    | = ***** GX  | 8-101 8-102 8-103 8-104   |
| CFGBF   | = ***** GX  | 9-127   |
| CFGSZ   | = ***** GX  | 9-126   |
| CFLIN   | = ***** GX  | 8-101 8-102 8-103 8-104   |
| CHKCUG  | = 000646 R  | 13-267 #14-335  |
| COUNT   | = ***** GX  | *12-210 *12-233 12-236 *12-244 12-246 12-248 13-269             |
| CUENC   | = 000406 R  | *13-267   |
| CUFG    | = 000364 R  | *13-262   |
| CUGDF   | = 000000 R  | 9-123   |
| CUGDIG  | = 000174 R  | *12-218   |
| CUGEND  | = 000011 R  | *7-92 12-218  |
| CUGFLG  | = 000014 R  | *7-94 *11-195 *13-266 *13-271 13-303                            |
| CUGKW   | = 000000 RG | 9-125 #10-142   |
| CUGNAM  | = 000000 R  | *7-89 11-185 11-190 13-289 14-348                               |
| CUGNLN  | = 000004 R  | *7-90 7-91 7-93 12-205 12-251                                   |
| CUGNUM  | = 000006 R  | *7-91 12-206 12-211 12-231 *12-231 *12-232 12-238 12-240 12-242 |
| CUGPCK  | = 000012 R  | *12-242 *12-243 12-252  |
| CUGST   | = 000000 RG | *7-93 12-253 13-296   |
| CUNAM   | = 000052 R  | #10-142   |
| CUNBEN  | = 000220 R  | #11-182   |
| CUNBST  | = 000136 R  | #12-229   |
| CUNMMX  | = ***** GX  | #12-205   |
| DTEDES  | = ***** GX  | 11-183 13-288 14-347  |
| FMT10   | = ***** GX  | 13-302 14-345   |
| FMT8    | = ***** GX  | 8-101 8-102 8-103   |
| FM.10   | = 000000    | 8-104   |
| FM.8    | = 000000    | #8-101 #8-102 #8-103  |
| GF\$BUG | = 000001    | #8-104  |
| G\$CUG  | = 000010    | 13-271 13-298   |
| G\$DTE  | = 000012    | *13-302 14-345  |
| G\$FLG  | = 000014    | *13-303   |
| G\$LEN  | = 000016    | 13-274  |
| G\$NAM  | = 000002    | 13-291 14-350   |
| H\$CUG  | = 000010    | 13-305 14-336   |
| KSARS   | = ***** GX  | 13-307 14-339 14-356  |
| LNKEND  | = ***** GX  | 13-306  |
| NEXT    | = ***** GX  | *12-211 12-218 14-355   |
| N\$VCT  | = *****     | 13-277  |
| PCKBCD  | = ***** GX  | 12-254  |
| PSIPT   | = ***** GX  | 13-304 14-335   |
| RTSPC   | = ***** GX  | 8-101 8-102 8-103 8-104   |
| SPACE   | = 000040    | *6-79 11-187  |
| STRNXT  | = ***** GX  | 12-212 12-220   |
| SYNERR  | = ***** GX  | *9-128 *9-131   |
| S\$BAS  | = *****     | 8-101 8-101 8-102 8-102 8-103 8-103 8-104 8-104                 |
| \$ALPHA | = 000022    | #10-142   |
| \$ANY   | = 000020    | #10-142   |
| \$BLANK | = 000006    | #10-142   |
| \$CEACX | = ***** GX  | 13-282 14-343   |
| \$DIGIT | = 000024    | #10-142   |
| \$DNUMB | = 000014    | #10-142   |

```

226      ;
227      ; LOW END OF SUBADDRESS RANGE
228      ;
229      000176      ;
230      000202      103405      DASSLO: CALL      CHKS      ; CHECK SUBADDRESS
231      000204      010167      BCS      101$      ; BR IF ILLEGAL
232      000210      010167      MOV      R1,DSASLO      ; STORE LOW SUBADDRESS RANGE
233      000214      C70000G      MOV      R1,DSASHI      ; DEFAULT HIGH SUBADDRESS IF OMITTED
234      ;
235      ; ERRORS
236      ;
237      000216      101$:      MSG$R      N9      ; ILLEGAL LOW SUBADDRESS
238      ;
239      ;
240      ; START OF HIGH SUBADDRESS
241      ;
242      ;
243      000224      ;
244      000230      103406      DASSHI: CALL      CHKS      ; CHECK SUBADDRESS
245      000232      020167      BCS      101$      ; BR IF ILLEGAL
246      000236      002403      CMP      R1,DSASLO      ; CHECK RANGE VALID
247      000240      010167      BLT      101$      ; BR IF ILLEGAL: HIGH LESS THAN LOW
248      000244      000000G      MOV      R1,DSASHI      ; STORE HIGH END OF SUBADDRESS RANGE
249      ;
250      ; ERRORS
251      ;
252      000246      101$:      MSG$R      M0      ; ILLEGAL HIGH SUBADDRESS
253      ;
254      ;
255      ; ROUTINE TO CHECK SUBADDRESS
256      ;
257      000254      005767      000000G      CHKS:      TST      ,PNUMH      ; TOO BIG?
258      000260      001010      BNE      20$      ; BR IF YES
259      000262      016701      000000G      MOV      ,PNUMB,R1      ; GET SUBADDRESS
260      000266      002405      BLT      20$      ; BR IF ILLEGAL
261      000270      020127      000000G      CMP      R1,#SUBAMX      ; NUMBER WITHIN LEGAL RANGE?
262      000274      101002      BHI      20$      ; BR IF ILLEGAL
263      000276      000241      CLC      ; ELSE SHOW NO ERROR
264      000300      000401      BR      30$      ; RETURN
265      000302      000261      20$:      SEC      ; FLAG ERROR
266      000304      30$:      RETURN      ;

```

100  
101  
102  
103  
104 000000  
105 000032  
106 000062  
107 000110  
108 000132  
109 000204  
110 000236  
111  
112  
113  
114  
115 000314

.SBTTL ERROR MESSAGES

.ENABL LC

NTLERS ,N4,10,CERR,RTSPC,CFLIN,<destination name>  
NTLERS ,N5,10,CERR,RTSPC,CFLIN,<priority number>  
NTLERS ,N6,10,CERR,RTSPC,CFLIN,<object number>  
NTLERS ,N7,10,CERR,RTSPC,CFLIN,<task name>  
NTLERS ,M,10,CERR,RTSPC,CFLIN,<call data mask/value combination>  
NTLERS ,M7,10,CERR,RTSPC,CFLIN,<destination name>  
NTLERS ,M9,8,CERR,RTSPC,CFLIN,<Destination block allocation failure>

.DSABL LC  
.LIST BEX  
.EVEN  
.PSECT



## SYMBOL CROSS REFERENCE

CREF 04.00

## SYMBOL VALUE REFERENCES

|        |             |         |         |         |        |        |        |       |
|--------|-------------|---------|---------|---------|--------|--------|--------|-------|
| ALLDST | = 001104 R  | 13-303  | #15-360 |         |        |        |        |       |
| BIAS   | = 000002    | #6-95   | #6-97   |         |        |        |        |       |
| CERR   | = ***** GX  | 7-104   | 7-105   | 7-106   | 7-107  | 7-108  | 7-109  | 7-110 |
| CFGBF  | = ***** GX  | 8-134   |         |         |        |        |        |       |
| CFGSZ  | = ***** GX  | 8-133   |         |         |        |        |        |       |
| CFLIN  | = ***** GX  | 7-104   | 7-105   | 7-106   | 7-107  | 7-108  | 7-109  | 7-110 |
| CHKDSN | = 001012 R  | 13-291  | #14-327 |         |        |        |        |       |
| CUNMMX | = ***** GX  | 10-213  | 15-395  |         |        |        |        |       |
| DSACUG | = ***** GX  | 10-214  | 15-394  |         |        |        |        |       |
| DSARCT | = ***** GX  | *10-224 | 15-381  |         |        |        |        |       |
| DSASHI | = ***** GX  | *10-221 | 15-391  |         |        |        |        |       |
| DSASLO | = ***** GX  | *10-220 | 15-390  |         |        |        |        |       |
| DSCMCT | = ***** GX  | *10-222 | 13-283  | 15-382  |        |        |        |       |
| DSCVCT | = ***** GX  | *10-223 | 13-283  |         |        |        |        |       |
| DSFLG  | = ***** GX  | *8-130  | 13-286  | 16-433  |        |        |        |       |
| DSNAM  | = 000372 R  | #10-199 |         |         |        |        |        |       |
| DSNAMX | = 000006    | #6-83   | 10-200  | 10-204  | 14-337 | 15-374 |        |       |
| DSOBJ  | = 000536 R  | #11-241 |         |         |        |        |        |       |
| DSPRI  | = 000530 R  | #11-236 |         |         |        |        |        |       |
| DSTDF  | = 000000 R  | 8-129   |         |         |        |        |        |       |
| DSTEND | = 000664 RG | #13-283 |         |         |        |        |        |       |
| DSTINS | = 001370 R  | 15-413  | #16-431 |         |        |        |        |       |
| DSTKMX | = 000006    | #6-84   | 12-261  |         |        |        |        |       |
| DSTKW  | = 000000 RG | 8-132   | #9-151  |         |        |        |        |       |
| DSTNAM | = ***** GX  | 10-202  | 14-336  | 15-377  |        |        |        |       |
| DSTOBJ | = ***** GX  | 11-241  | 11-251  | 15-387  |        |        |        |       |
| DSTPRI | = ***** GX  | 11-236  | 15-386  | 16-444  |        |        |        |       |
| DSTSK  | = 000606 R  | #12-261 |         |         |        |        |        |       |
| DSTST  | = 000000 RG | #9-151  |         |         |        |        |        |       |
| DSTTSK | = ***** GX  | *10-218 | *10-219 | 12-264  | 15-388 | 15-389 |        |       |
| DSTVAR | = ***** GX  | 13-294  | 15-400  |         |        |        |        |       |
| DS.X29 | = ***** GX  | 13-286  | 16-433  |         |        |        |        |       |
| DSUGN  | = 000022    | 15-393  |         |         |        |        |        |       |
| DSOATL | = 000030    | *15-385 |         |         |        |        |        |       |
| DSOST  | = 000010    | 15-376  |         |         |        |        |        |       |
| DSOTEL | = 000031    | *15-381 |         |         |        |        |        |       |
| DSFLEN | = 000032    | 13-293  |         |         |        |        |        |       |
| DSNAM  | = 000004    | 14-335  | *15-388 | *15-389 |        |        |        |       |
| DSOBJ  | = 000003    | *15-387 |         |         |        |        |        |       |
| DSPRI  | = 000002    | *15-386 | 16-444  |         |        |        |        |       |
| DSHI   | = 000020    | *15-391 |         |         |        |        |        |       |
| DSLO   | = 000016    | *15-390 |         |         |        |        |        |       |
| DSVBL  | = 000032    | *15-399 |         |         |        |        |        |       |
| FMT10  | = ***** GX  | 7-104   | 7-105   | 7-106   | 7-107  | 7-108  | 7-109  |       |
| FMT8   | = ***** GX  | 7-110   |         |         |        |        |        |       |
| FM.10  | = 000000    | #7-104  | #7-105  | #7-106  | #7-107 | #7-108 | #7-109 |       |
| FM.8   | = 000000    | #7-110  |         |         |        |        |        |       |
| HSDST  | = 000012    | 13-288  | 16-432  | 16-435  |        |        |        |       |
| HSD29  | = 000014    | 13-290  | 16-435  |         |        |        |        |       |
| KSAR5  | = ***** GX  | 14-328  | 14-343  | 15-365  | 15-411 |        |        |       |
| MSMGF  | = 000000    | 6-96    |         |         |        |        |        |       |
| MSVCT  | = *****     | 14-342  | 15-363  |         |        |        |        |       |

```

239          .SBTTL DTE$DF ACTION ROUTINES
240          ;
241          ; START OF DTE ADDRESS
242          ;
243 000052 005067 000000G DTEAST: CLR DTEFLG ; INITIALISE FLAGS (ON/OFF BITS)
244 000056 012700 000017 MOV #DTEALN,R0 ; GET LENGTH OF DTE ADDRESS
245 000062 012701 000000' MOV #DTEADD,R1 ; GET ADDRESS OF BUFFER
246 000066 105021 10$: CLR (R1)+ ; INITIALIZE BUFFER
247 000070 005300 DEC R0 ; MORE TO INITIALIZE?
248 000072 003375 BGT 10$ ; BR IF YES
249 000074 105067 000000G CLR COUNT ; INITIALIZE DTE DIGIT COUNT
250 000100 012767 000000' 000000G MOV #DTEADD,NEXT ; START STORING AT BEGINNING OF BUFFER
251 000106 CALL STRNXT ; STORE CHARACTER
252 000112 RETURN
253
254          ;
255          ; PROCESS DTE ADDRESS DIGIT
256          ;
257 000114 026727 000000G 000016' DTEDIG: CMP NEXT,#DTEND ; IS ADDRESS TOO MANY CHARACTERS?
258 000122 101003 BHJ 101$ ; BR IF YES
259 000124 CALL STRNXT ; STORE NEXT DIGIT
260 000130 RETURN
261
262          ;
263          ; ERRORS
264 000132 101$: MSG$R 03 ; ILLEGAL DTE ADDRESS
265
266          ;
267          ; END OF DTE ADDRESS
268          ;
269 000140 004567 000000G DTEAEN: JSR R5,$$AVRG ; SAVE R3-R5
270 000144 012700 000010 MOV #<DTEALN+1>/2,R0 ; GET LENGTH OF PACKED ADDRESS
271 000150 012704 000000' MOV #DTEADD,R4 ; GET ADDRESS OF DTE ADDRESS BUFFER
272 000154 012705 000017' MOV #DTEPCK,R5 ; GET ADDRESS OF PACKED ADDRESS BUFFER
273 000160 CALL PCKBCD ; PACK ADDRESS IN BCD FORMAT
274
275 000164 017700 000000G MOV @PSIPT,R0 ; POINT TO HOME BLOCK
276 000170 062700 000024 ADD #H$NETW,R0 ; POINT TO DEFAULT NET NAME
277 000174 012701 000027' MOV #DTNINM,R1 ; POINT TO LOCAL STORAGE OF NAME
278 000200 012702 000006 MOV #DTNMMX,R2 ; GET SIZE OF NAME
279 000204 112021 10$: MOV (R0)+,(R1)+ ; SET UP DEFAULT FOR DTE
280 000206 005302 DEC R2 ; MORE TO SET UP?
281 000210 003375 BGT 10$ ; YES, SET AWAY
282 000212 RETURN ; ELSE COMPLETED

```

## SYMBOL CROSS REFERENCE

CREF    04.00

| SYMBOL    | VALUE        | REFERENCES                 |
|-----------|--------------|----------------------------|
| \$HEADR   | = ***** GX   | 15-355    16-459    17-512 |
| \$LAMDA   | = 000000     | #10-168                    |
| \$NUMBR   | = 000002     | #10-168                    |
| \$QDTE    | = 000000 RG  | #9-147                     |
| \$RAD50   | = 000016     | #10-168                    |
| \$RONLY   | = *****      | 10-168    10-168           |
| \$SAVRG   | = ***** GX   | 12-269                     |
| \$STRNG   | = 000004     | #10-168                    |
| \$SUBXP   | = 000010     | #10-168                    |
| \$XALOC   | = * ***** GX | 15-353                     |
| \$\$\$FLG | = 177777     | #10-168                    |
| \$\$\$KEY | = 177777     | #10-168                    |
| .PNUMB    | = ***** GX   | 13-289    13-309           |
| .PNUMH    | = ***** GX   | 13-287    13-307           |
| .PSTCN    | = ***** GX   | 14-321                     |
| .PSTPT    | = ***** GX   | 14-324                     |
| .TPARS    | = ***** GX   | 9-153                      |

```

269
270
271      ; LOW END OF SUBADDRESS RANGE
272
273      PSNSLO: CALL    CHKSUB          ; CHECK FOR VALID SUBADDRESS VALUE
274      BCS          101$             ; BR IF INVALID VALUE
275      MOV          R1,PSNLO         ; SAVE SUBADDRESS VALUE
276      MOV          R1,PSNHI         ; DEFAULT IF HIGH OMITTED
277      BIS          #HF$DLM,PSNFLG   ; INDICATE SUBADDRESS WAS SPECIFIED
278      RETURN
279
280      ; ERRORS
281
282      101$:  EMSG$R  D1              ; INVALID LOW SUBADDRESS VALUE
283
284      ; CHECK FOR VALID SUBADDRESS VALUE
285
286      CHKSUB: TST      .PNUMH        ; LEGAL VALUE
287      BNE          10$              ; BR IF ILLEGAL VALUE
288      MOV          .PNUMB,R1         ; GET SUBADDRESS VALUE
289      BLT          10$              ; BR IF ILLEGAL VALUE
290      CMP          R1,#SUBAMX        ; LEGAL VALUE?
291      BHI          10$              ; BR IF NO
292      CLC
293      BR           20$              ; INDICATE LEGAL VALUE
294      SEC
295      AND EXIT
296      RETURN
297
298      10$:  SEC
299      20$:  RETURN

```

```
152 .SBTTL ERROR MESSAGES
153
154 .ENABL LC
155
156 000040 NTLERS ,NO,10,CERR,RTSPC,CFLIN,<circuit identification>
157 000100 NTLERS ,N1,10,CERR,RTSPC,CFLIN,<process owner name>
158 000134 NTLERS ,N2,10,CERR,RTSPC,CFLIN,<flags byte value>
159 000166 NTLERS ,N3,8,CERR,RTSPC,CFLIN,<Resource allocation failure>
160
161 .DSABL LC
162 .EVEN
163 .LIST BEX
164 000000 .PSECT
```

```

661 .SBTTL PVDLM - CHECK FOR DLM PVC
662
663
664
665 PVDLM - CHECK FOR DLM PVC
666
667 INPUTS:
668 PVCNAM - PROCESS NAME (ASCII)
669
670 OUTPUTS:
671 DLMFV - DLM'S PDV ADDRESS
672
673
674 001446 012700 000024' PVDLM: MOV #PVCNAM,R0 ; POINT TO SPECIFIED PROCESS NAME
675 001452 CALL $CAT5 ; CONVERT TO RAD50
676 001456 020127 015355 CMP R1,#ARDLM ; IS THIS DLM?
677 001462 001025 BNE INVCIR ; BR IF NO - ILLEGAL CIRCUIT ID
678 001464 010102 MOV R1,R2 ; GET DLM'S RAD50 PROCESS NAME
679
680 001466 SWSTK$ 20$ ; ENTER SYSTEM STATE
681 001472 CALL @PDVID ; GET DLM'S PDV INDEX
682 001476 103006 BEC 10$ ; BR IF FOUND IT
683 001500 RETC R0 ; ELSE SET USER C-BIT
684 001512 000406 BR 15$ ; AND EXIT
685 001514 110267 000035' 10$: MOV R2,DLMIND ; SAVE PDV INDEX
686 001520 067702 000000G ADD @PDVTA,R2 ; POINT TO ENTRY IN PDV VECTOR TABLE
687 001524 011267 000014' MOV (R2),DLMFV ; SAVE DLM'S PDV INDEX
688 001530 15$: RETURN ; EXIT SYSTEM STATE
689
690 001532 103401 20$: BCS INVCIR ; BR IF DLM NOT IN SYSTEM
691 001534 RETURN
692
693
694
695 001536 INVCIR: MSG$R NO ; ILLEGAL CIRCUIT ID
696
697
698
699
700 001544 116767 000000G 000032' PVCTL: .ENABL LSB
701 001552 000403 BR .PNUMB,DLMCTL ; SAVE CONTROLLER NUMBER
702 001554 116767 000000G 000034' PVSTA: MOV .PNUMB,DLMSTA ; CHECK FOR VALIDITY
703 001562 005767 000000G 10$: TST .PNUMB ; SAVE STATION NUMBER
704 001566 001363 BNE INVCIR ; LEGAL VALUE?
705 001570 105767 000001G TST .PNUMB+1 ; BR IF NO - ERROR
706 001574 001360 BNE INVCIR ; LEGAL VALUE?
707 001576 RETURN ; BR IF NO
708 .DSABL LSB
709
710 000001 .END

```

```
99          .SBTTL  ERROR MESSAGES
100
101          .ENABL  LC
102
103 000036    NTLERS$ ,L0,10,CERR,RTSPC,CFLIN,<remote dte name>
104 000066    NTLERS$ ,L1,10,CERR,RTSPC,CFLIN,<remote dte address>
105 000122    NTLERS$ ,L2,8,CERR,RTSPC,CFLIN,<Remote dte block allocation failure>
106
107          .DSABL  LC
108          .LIST   BEX
109          .EVEN
110 000000    .PSECT
```

```
52 .SBTTL MACRO DEFINITIONS
53
54 ;
55 ; LIBRARY MACROS
56 ;
57 .MCALL MSG$,NTLR$,X29DF$,PHBDF$,ISTAT$,STATE$,TRANS
58
59 000000 X29DF$ ; X.29 DATA BLOCK DEFINITIONS
60 000000 PHBDF$ ; PSI HOME BLOCK DEFINITIONS
61
```



77  
78  
79  
80  
81  
82

.SBTTL LOCAL SYMBOL DEFINITIONS

:: LOCAL SYMBOL DEFINITIONS

000001

CHNLN = 1.

; MINIMUM VALUE FOR CHANNEL RANGE

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51

.TITLE CTBIO - CETAB LOADING ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NLT - CETAB LOADING ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

ELTIME - CALCULATE ELAPSED TIME MACRO V05.03b Saturday 29-Jun-85 17:42 Page 5

```

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73 000000
74 000000 017746 000000G
75 000004 160016
76 000006 103002
77 000010 062716 124300
78 000014 000241
79 000016 012600
80 000020
81
82
83 000001

.SBTTL ELTIME - CALCULATE ELAPSED TIME
;+
;*** - ELTIME - CALCULATE ELAPSED TIME
;
;INPUTS:
;RO = START TIME GIVEN IN NUMBER OF 2 SECOND INTERVALS SINCE MIDNIGHT
;
;OUTPUT:
;C-BIT = SUCCESS/FAILURE
;RO = TIME ELAPSED FROM START TIME TO CURRENT TIME. RESULT IS AN UNSIGNED
;16-BIT INTEGER.
;
;ALL OTHER REGISTERS PRESERVED.
;-

ELTIME::
MOV @ZTJM2, -(SP) ; GET TIME IN 2 TICK UNITS ; RJK01
SUB RO, (SP) ; CALCULATE ELAPSED TIME
BCC 10$ ; IF CC, CURRENT TIME > START TIME
ADD #43200., (SP) ; ELSE, ADD 24 HOURS TO RESULT
10$: CLC ; DON'T CORRUPT THE C-BIT!
MOV (SP)+, RO ; RESULT RETURNED IN RO
20$: RETURN

.END

```

LBNIO - LOGICAL BLOCK I/O ROUTE MACRO V05.03b Saturday 29-Jun-85 00:21 Page 10-1

```

Symbol table
A$$CHK= 000000      G$$TSS= 000000      L$$P11= 000001      N$$NCT= 000001      T$$MIN= 000000
A$$CPS= 000000      G$$ITK= 000000      L$$11R= 000000      N$$PEM= 000001      UISAV= 000000R      002
A$$PRI= 000000      G$$WRD= 000000      M$$CRB= 000124      PR7 = ***** GX      USARO = ***** GX
A$$TRP= 000000      IO_RLB= ***** GX      M$$CRX= 000000      P$$P45= 000000      V$$CTR= 001000
C$$CKP= 000000      I$$RAR= 000000      M$$FCS= 000000      P$$WRD= 000000      X$$DBT= 000000
C$$ORE= 000400      I$$RDN= 000000      M$$MGE= 000000      Q$$OPT= 000010      $CLOPE 000000RG
C$$RSH= 177564      K$$CNT= 177546      M$$NET= 000000      R$$DER= 000000      $CLQST= ***** GX
D$$BUG= 177514      K$$CSR= 177546      M$$OVR= 000000      R$$K11= 000001      $IOSB 000044RG      002
D$$ISK= 000000      K$$LDC= 000000      N$$ACC= 000001      R$$SND= 000000      $LBN 000036RG      002
D$$L11= 000001      K$$TPS= 000074      N$$BUF= 000001      R$$11M= 000000      $LBUF 000030RG      002
D$$SYNC= 000000      LBNDPB 000014R      002 N$$LDV= 000001      STAT 000002R      002 $LEN 000032RG
D$$SYNM= 000000      LD$LP = 000000      N$$MCP= 000001      S$$WRG= 000700      $READ 000157RG
E$$XPR= 000000      L$BHRB= ***** GX      N$$MLL= 000001      S$$YSZ= 007600      $READX 000060RG
F$$LVL= 000001      L$$ASG= 000000      N$$MOV= 000010      T$$KMG= 000000      $RLBL 000026RG
G$$TPP= 000000      L$$DRV= 000000

```

```

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)
        000200 001 (RW,I,LCL,REL,CON)
DATA 000050 002 (RW,D,LCL,REL,CON)
Errors detected: 0

```

\*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 9100 Words ( 36 Pages)
Size of core pool: 14440 Words ( 55 Pages)
Operating system: RSX-11M/PLUS

```

```

Elapsed time: 00:00:08.33
SY: LBNIO.V2,[132,134]LBNIO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]LBNIO

```

```
52                                     .SBTTL  MACRO DEFINITIONS
53
54                                     ;
55                                     ; LIBRARY MACROS
56                                     ;
57                                     .MCALL  CUGDF$,RETC,EMSG$R,PHBDF$,NTLERS$,STATES$,TRANS$,ISTAT$
58
59 000000                                CUGDF$                                ; CLOSED USER GROUP BLOCK OFFSETS
60 000000                                PHBDF$                                ; PSI HOME BLOCK OFFSETS
61
62                                     ;
63                                     ; LOCAL MACRO DEFINITIONS
64                                     ;
65                                     .MACRO  SAVMAP
66 MOV    @KSAR5,-(SP)                  ; SAVE APR 5
67 .ENDM
68
69                                     .MACRO  RESMAP
70 MOV    (SP)+,@KSAR5                  ; RESTORE APR5
71 .ENDM
72
```

CFPCUG      CREATED BY MACRO ON 29-JUN-85 AT 00:15      PAGE 2      E 2

SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL    | VALUE      | REFERENCES           |
|-----------|------------|----------------------|
| \$EOS     | = 000012   | #10-142              |
| \$ERRM1   | 000110 R   | #8-104 13-317        |
| \$ERRN2   | 000016 R   | #8-101 11-200 13-316 |
| \$ERRN3   | 000040 R   | #8-102 12-225        |
| \$ERRP4   | 000064 R   | #8-103 13-315        |
| \$ERR1T   | = ***** GX | 9-138                |
| \$EXIT    | = 000000   | #10-142              |
| \$FAIL    | = 177777   | #10-142              |
| \$CPRM    | = *****    | 10-142               |
| \$HEADR   | = ***** GX | 13-277 14-355        |
| \$LAMDA   | = 000000   | #10-142              |
| \$NUMBR   | = 000002   | #10-142              |
| \$QCUG    | 000000 RG  | #9-123               |
| \$RAD50   | = 000016   | #10-142              |
| \$RONLY   | = *****    | 10-142 10-142        |
| \$SAVRG   | = ***** GX | 12-250               |
| \$STRNG   | = 000004   | #10-142              |
| \$SUBXP   | = 000010   | #10-142              |
| \$XALOC   | = ***** GX | 13-275               |
| \$\$\$FLG | = 177777   | #10-142              |
| \$\$\$KEY | = 177777   | #10-142              |
| .PNUMB    | = ***** GX | 13-264 13-266        |
| .PNUMH    | = ***** GX | 13-262               |
| .PSTCN    | = ***** GX | 11-182               |
| .PSTPT    | = ***** GX | 11-191               |
| .TPARS    | = ***** GX | 9-129                |

```

268
269
270
271
272 000306 016702 000000G
273 000312 022702 000000G
274 000316 103410
275 000320 012701 000000G
276 000324 016700 000000G
277 000330 112021
278 000332 005302
279 000334 003375
280 000336
281
282
283
284 000340

      ; CLOSED USER GROUP NAME
DACUNM: MOV      ,PSTCN,R2
        CMP      #CUNMMX,R2
        BLO      101$
        MOV      #DSACUG,R1
        MOV      ,PSTPT,R0
10$:    MOV      (R0)+,(R1)+
        DEC      R2
        BGT      10$
        RETURN

      ; ERRORS
101$:   MSG$R N2
      ; ILLEGAL CUG NAME

      ; GET NUMBER OF CHARACTERS SPECIFIED
      ; LEGAL NUMBER OF CHARACTERS?
      ; BR IF NO
      ; GET ADDRESS OF CUG NAME STORAGE
      ; GET ADDRESS OF SPECIFIED NAME
      ; STORE CHARACTER
      ; MORE TO STORE?
      ; BR IF YES

```

```

117                                     .SBTTL LOOK FOR DST$DF MACRO
118
119
120                                     ;+
121                                     ;$QDST - LOOK FOR DST$DF MACRO
122                                     ;
123                                     ; INPUTS:
124                                     ;     NONE
125                                     ;
126                                     ; OUTPUTS:
127                                     ;     C-BIT=SUCCESS/FAILURE
128                                     ;     R3,R4,R5=DESTROYED
129                                     ;-
129 000314 012705 000000' $QDST:: MOV #DSTDF,R5 ; STATE TABLE ADDRESS
130 000320 005067 000000G CLR DSFLG ; INITIALIZE FLAGS WORD
131 000324 005001 CLR R1 ; FULL KEYWORD MATCH LENGTH
132 000326 012702 000000' MOV #DSTKW,R2 ; KEYWORD TABLE ADDRESS
133 000332 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
134 000336 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
135 000342 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
136 000346 CALL TPARS ; GO DO THE PARSE
137 000352 103003 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
138 000354 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
139 000360 001401 BEQ 101$ ; IF EQ, YES
140 000362 20$: RETURN
141
142                                     ;
143                                     ; ERROR CONDITION
144                                     ;
145 000364 101$: EMG$R 1T ; SYNTAX ERROR
    
```



| SYMBOL   | VALUE       | REFERENCES                                |
|----------|-------------|---|
| PSIPT    | = ***** GX  | 13-285 16-431                             |
| RTSPC    | = ***** GX  | 7-104 7-105 7-106 7-107 7-108 7-109 7-110 |
| R\$R0    | = 000002    | #6-88                                     |
| R\$R1    | = 000004    | #6-89                                     |
| R\$R2    | = 000006    | #6-90                                     |
| R\$R3    | = 000010    | #6-91                                     |
| R\$R4    | = 000012    | #6-92                                     |
| R\$R5    | = 000014    | #6-93                                     |
| SPACE    | = 000040    | #6-82 10-205 10-215                       |
| SYNERR   | = ***** GX  | *8-138                                    |
| \$B\$BAS | = *****     | 7-104 7-105 7-106 7-107 7-108             |
|          |             | 7-109 7-109 7-110 7-110                   |
| \$ALPHA  | = 000022    | #9-151                                    |
| \$ANY    | = 000020    | #9-151                                    |
| \$BLANK  | = 000006    | #9-151                                    |
| \$CAT5   | = ***** GX  | 12-267                                    |
| \$CEACX  | = ***** GX  | 14-332 15-368 16-442                      |
| \$DIGIT  | = 000024    | #9-151                                    |
| \$DNUMB  | = 000014    | #9-151                                    |
| \$EOS    | = 000012    | #9-151                                    |
| \$ERRM7  | = 000204 R  | #7-109 13-309                             |
| \$ERRM9  | = 000236 R  | #7-110 13-310                             |
| \$ERRN4  | = 000000 R  | #7-104 10-229                             |
| \$ERRN5  | = 000032 R  | #7-105 11-253                             |
| \$ERRN6  | = 000062 R  | #7-106 11-254                             |
| \$ERRN7  | = 000110 R  | #7-107 12-276                             |
| \$ERR1T  | = ***** GX  | 8-145                                     |
| \$ERR6M  | = 000132 R  | #7-108 13-311                             |
| \$EXIT   | = 000000    | #9-151                                    |
| \$FAIL   | = 177777    | #9-151                                    |
| \$GPRM   | = *****     | 9-151                                     |
| \$HEADR  | = ***** GX  | 14-342 15-363                             |
| \$LAMDA  | = 000000    | #9-151                                    |
| \$LINKX  | = ***** GX  | 16-449                                    |
| \$NUMBR  | = 000002    | #9-151                                    |
| \$QDST   | = 000314 RG | #8-129                                    |
| \$RAD50  | = 000016    | #9-151                                    |
| \$RONLY  | = *****     | 9-151 9-151                               |
| \$STRNG  | = 000004    | #9-151                                    |
| \$SUBXP  | = 000010    | #9-151                                    |
| \$XALOC  | = ***** GX  | 15-361                                    |
| \$S\$FLG | = 177777    | #9-151                                    |
| \$S\$KEY | = 177777    | #9-151                                    |
| .PNUMB   | = ***** GX  | 11-244 11-246                             |
| .PNUMH   | = ***** GX  | 11-242                                    |
| .PSTCN   | = ***** GX  | 10-199 12-261 *12-269                     |
| .PSTPT   | = ***** GX  | 10-209 12-265                             |
| .TPARS   | = ***** GX  | 8-136                                     |

```

284                                     ;
285                                     ; HASH TABLE SIZE
286                                     ;
287 000214 005767 000000G HSHTBL: TST .PNUMH ; LEGAL HASH TABLE SIZE VALUE?
288 000220 001016 ; BNE 101$ ; BR IF NO
289 000222 016700 000000G ; MOV .PNUMB,R0 ; GET HASH TABLE SIZE
290 000226 012701 000040 ; MOV #HSHMN,R1 ; GET MINIMUM SIZE OF HASH TABLE
291 000232 020001 10$: CMP R0,R1 ; IS THIS A LEGAL SIZE?
292 000234 001405 ; BEQ 20$ ; BR IF YES
293 000236 006301 ; ASL R1 ; HASH TABLE SIZE MUST BE POWER OF TWO
294 000240 020127 001000 ; CMP R1,#HSHMX ; IS THIS A LEGAL SIZE?
295 000244 101004 ; BHI 101$ ; BR IF NO
296 000246 000771 ; BR 10$ ; CHECK FOR LEGAL VALUE
297 000250 010067 000000G 20$: MOV R0,HSHGZ ; SAVE HASH TABLE SIZE
298 000254 ; RETURN
299 ;
300 ; ERRORS
301 ;
302 000256 101$: MSG$R 04 ; ILLEGAL HASH TABLE SIZE
303 ;
304 ;
305 ; COUNTER TIMER VALUE
306 ;
307 000264 005767 000000G CNTTIM: TST .PNUMH ; LEGAL COUNTER TIMER VALUE?
308 000270 001004 ; BNE 101$ ; BR IF NO
309 000272 016767 000000G 000000G ; MOV .PNUMB,CTIM ; SAVE COUNTER TIMER VALUE
310 000300 ; RETURN
311 ;
312 ; ERRORS
313 ;
314 000302 101$: MSG$R 05 ; ILLEGAL COUNTER TIMER VALUE
315 ;

```

CFPDTE CREATED BY MACRO ON 29-JUN-85 AT 00:17 PAGE 5 E 7

MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES |         |         |         |         |         |         |         |         |         |
|------------|------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| CALL       | 9-153      | 12-251  | 12-259  | 12-273  | 15-340  | 15-344  | 15-352  | 15-353  | 15-361  | 15-403  |
|            | 15-409     | 16-442  | 16-449  | 17-485  | 17-490  |         |         |         |         |         |
| DBGTP\$    | #10-168    | #10-180 | #10-199 | #10-203 | #10-207 | #10-214 | #11-232 |         |         |         |
| DTEDF\$    | #5-60      | 5-65    |         |         |         |         |         |         |         |         |
| EMSG\$R    | #5-60      | 9-162   | 12-264  | 13-302  | 13-314  | 14-334  | 15-418  | 15-419  | 15-420  | 15-421  |
| ISTAT\$    | #5-61      | 10-168  |         |         |         |         |         |         |         |         |
| MTRANS\$   | #10-168    |         |         |         |         |         |         |         |         |         |
| NTLRS\$    | #5-61      | 8-124   | 8-125   | 8-126   | 8-127   | 8-128   | 8-129   |         |         |         |
| PHBDF\$    | #5-61      | 5-66    |         |         |         |         |         |         |         |         |
| RESMAP     | #5-75      | 15-410  | 16-460  |         |         |         |         |         |         |         |
| RESRG      | #5-60      |         |         |         |         |         |         |         |         |         |
| RETC       | #5-60      | 15-355  | 16-459  | 17-512  |         |         |         |         |         |         |
| RETURN     | 9-157      | 12-252  | 12-260  | 12-282  | 13-298  | 13-310  | 15-411  | 15-414  | 16-462  | 17-514  |
|            | 18-535     |         |         |         |         |         |         |         |         |         |
| SAVMAP     | #5-71      | 15-358  | 16-443  |         |         |         |         |         |         |         |
| SAVRG      | #5-60      |         |         |         |         |         |         |         |         |         |
| SLTDF\$    | #5-60      | 5-67    |         |         |         |         |         |         |         |         |
| STATE\$    | #5-61      | 10-172  | #10-175 | #10-178 | #10-182 | #10-185 | #10-188 | #10-191 | #10-194 | #10-197 |
|            | #10-201    | #10-205 | #10-209 | #10-212 | 11-221  | #11-224 | #11-227 | #11-230 | #11-234 | #11-237 |
| SWSTK\$    | 15-352     | 16-442  | 17-485  |         |         |         |         |         |         |         |
| TRANS      | #5-61      | #10-173 | #10-176 | #10-179 | #10-180 | #10-183 | #10-186 | #10-189 | #10-192 | #10-195 |
|            | #10-198    | #10-199 | #10-202 | #10-203 | #10-206 | #10-207 | #10-210 | #10-213 | #10-214 | #11-222 |
|            | #11-225    | #11-228 | #11-231 | #11-232 | #11-235 |         |         |         |         |         |

```

299      :
300      : HIGH END OF SUBADDRESS RANGE
301      :
302      :
303      PSNSHI: .ENABL  LSB
304      000330      CALL  CHKSUB      ; CHECK FOR VALID SUBADDRESS VALUE
305      000334      BCS   101$      ; BR IF INVALID VALUE
306      000336      CMP   R1,PSNLO  ; HIGH MUST NOT BE LESS THAN LOW
307      000340      BLT   101$      ; BR IF ERROR
308      000344      MOV   R1,PSNHI  ; SAVE SUBADDRESS VALUE
309      000350      BIS   #HF$DLM,PSNFLG ; INDICATE SUBADDRESS WAS SPECIFIED
310      :
311      : END OF PSN$DF
312      PSNEND :MOV   #H$LEN,R1      ; GET LENGTH OF PSI HOME BLOCK
313      000362      CALL  $ALL16     ; ALLOCATE HOME BLOCK FROM DSR
314      000366      BCS   102$      ; BR IF UNABLE TO ALLOCATE IT
315      000370      SAVRG <R0>      ; SAVE HOME BLOCK ADDRESS
316      000372      MOV   R0,R2     ; GET ADDRESS OF HOME BLOCK
317      000374      MOV   #H$LEN,R1 ; GET LENGTH OF HOME BLOCK
318      000376      5$: CLR B (R2)+ ; INITIALIZE HOME BLOCK
319      000380      DEC   R1         ; MORE TO INITIALIZE?
320      000384      BGT   5$        ; BR IF YES
321      000388      ADD   #H$NFTW,R0 ; POINT TO NETWORK NAME IN HOME BLOCK
322      000392      MOV   #PSNNAM,R1 ; POINT TO SAVED NETWORK NAME
323      000396      MOV   #PNAMM,R2  ; GET LENGTH OF NETWORK NAME
324      000400      10$: MOVB (R1)+,<R0>+ ; STORE NAME (ALREADY BLANK FILLED)
325      000404      DEC   R2        ; MORE TO STORE?
326      000408      BGT   10$       ; BR IF YES
327      000412      MOV   PORTNO,R1 ; GET NUMBER OF PORTS
328      000416      ASL   R1        ; GET NUMBER OF BYTES TO ALLOCATE
329      000420      RESRG <R0>      ; RETRIEVE HOME BLOCK ADDRESS
330      000424      MOV   R0,@PSIPT ; STORE HOME BLOCK ADDRESS
331      :
332      000444      SWSTK$ 50$      ; ENTER SYSTEM STATE
333      000450      CALL  $XALOC     ; ALLOCATE PORT TABLE FROM NETWORK POOL
334      000454      BCC   20$      ; BR IF SUCCESSFUL ALLOCATION
335      000458      RETC   R0       ; SET USER C-BIT
336      000462      BR    40$      ; AND EXIT
337      000466      20$: MOV   R0,R$R2(SP) ; RETURN PORT TABLE ADDRESS IN USER R2
338      000470      SAVMAP ; SAVE CURRENT MAPPING
339      000474      MOV   R0,-(SP)   ; GET UNMAPPED ADDRESS
340      000478      CALL  $CEACK    ; CONVERT TO MAPPED ADDRESS
341      000482      MOV   (SP)+,R2  ; RETRIEVE MAPPED ADDRESS
342      000486      MOV   PORTNO,R1 ; GET NUMBER OF WORDS IN PORT TABLE
343      000490      30$: CLR B (R2)+ ; INITIALIZE PORT TABLE
344      000494      DEC   R1        ; MORE TO INITIALIZE?
345      000498      BGT   30$       ; BR IF YES
346      000502      RESMAP ; RESTORE PREVIOUS MAPPING
347      000506      RETURN        ; RETURN TO USER STATE
348      :
349      000532      50$: BCS   103$  ; BR IF ERROR
350      000536      MOV   R2,H$PTB(R0) ; STORE PORT TABLE ADDR IN HOME BLOCK
351      000540      MOV   PORTNO,H$NPT(R0) ; STORE NUMBER OF PORTS
352      000544      MOV   PSNFLG,H$FLG(R0) ; STORE FLAGS WORD IN HOME BLOCK
353      000548      BIT   #HF$DLM,PSNFLG ; IS SUBADDRESS RANGE PRESENT?
354      000552      BEQ   60$      ; IF EQ NO
355      000556      MOV   PSNLO,H$LOTS(R0) ; FILL IN LOW TRANSPORT SUBADDRESS

```

```

166                                     ,SBTTL LOOK FOR PVC$DF MACRO
167
168                                     ;+
169                                     $QPVC - LOOK FOR PVC$DF MACRO
170                                     :
171                                     INPUTS:
172                                     NONE
173                                     :
174                                     OUTPUTS:
175                                     C-BIT=SUCCESS/FAILURE
176                                     R3,R4,R5=DESTROYED
177                                     :
178 000000 005067 000020' $QPVC:: CLR FLAG ; CLEAR LOCAL FLAGS WORD
179 000004 012705 000000' MOV #PVCDF,R5 ; STATE TABLE ADDRESS
180 000010 005001 CLR R1 ; FULL KEYWORD MATCH LENGTH
181 000012 012702 000000' MOV #PCVKW,R2 ; KEYWORD TABLE ADDRESS
182 000016 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
183 000022 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
184 000026 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
185 000032 CALL TPARS ; GO DO THE PARSE
186 000036 103003 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
187 000040 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
188 000044 001401 BEQ 101$ ; IF EQ, YES
189 000046
190 20$: RETURN
191
192                                     ; ERROR CONDITION
193
194 000050 101$: MSG$R 1T ; SYNTAX ERROR

```

|                   |                       |                  |                  |                   |
|-------------------|-----------------------|------------------|------------------|-------------------|
| ALLXCB 000640R    | G\$STTK= 000000       | M\$SNET= 000000  | R\$SSND= 000000  | X\$RPS 000012     |
| A\$SCHK= 000000   | G\$SWRD= 000000       | M\$SOVR= 000000  | R\$S11M= 000000  | X\$RTRY 000002    |
| A\$SCPS= 000000   | HF\$DLM= 000002       | N\$SACC= 000001  | SPACE = 00004C   | X\$RX1 000104     |
| A\$SPRI= 000000   | HF\$GWY= 000010       | N\$SBUF= 000001  | SP\$P4 000036RG  | X\$RXQ 000072     |
| A\$STRP= 000000   | HF\$HOS= 000004       | N\$SLDV= 000001  | SYNERR= ***** GX | X\$SS 000004      |
| BIASX 001426R     | HF\$XDF= 000020       | N\$SMCP= 000001  | S\$SWRG= 000000  | X\$ST 000005      |
| BLK\$ZD= ***** GX | H\$HADD= ***** GX     | N\$SMLL= 000001  | S\$SYSZ= 007600  | X\$STCLZ 000030   |
| CERR = ***** GX   | H\$HSH= ***** GX      | N\$SMOV= 000010  | T\$SKMG= 000000  | X\$STIMC 000001   |
| CFGBF = ***** GX  | H\$CUG = 000010       | N\$SNCT= 000001  | T\$SMIN= 000000  | X\$STIMR 000000   |
| CFGSZ = ***** GX  | H\$DST 000012         | N\$SPEM= 000001  | V\$CTR= 001000   | X\$STI 000102     |
| CFLIN = ***** GX  | H\$D29 000014         | OWNER 000112R    | W\$DSZ 000070R   | X\$STXQ 000066    |
| CF\$BLK= 000102   | H\$FLG 000000         | PCVKW 000000RG   | W\$DZD= ***** GX | X\$STYP 000015    |
| CHKBLK= ***** GX  | H\$GLEN 000104        | PDVID = ***** GX | X\$SNCO= 000020  | X\$USR 000022     |
| CHKDLM 000166R    | H\$GLT 000044         | PDVTA = ***** GX | X\$SRR = 000200  | X\$SWAQ 000062    |
| CHKPVC 000742R    | H\$GNAM 000050        | PFLAGS 000134R   | X\$SUCL= 000040  | X\$WSZ 000020     |
| CHKWND= ***** GX  | H\$GNML= 000020       | PF.DLM= 000001   | X\$SURE= 000100  | X\$SDBT= 000000   |
| CHNLMX= ***** GX  | H\$GPT 000046         | PORTNO= ***** GX | X\$SNCO= 000020  | ZTIME = ***** GX  |
| CHNLNO 000022R    | H\$HITS 000034        | PRTADD 000000R   | X\$SOFF= 000200  | ALPHA= 000022     |
| \$DTE 000012      | H\$HLEN 000044        | PSIPT = ***** GX | X\$SORI= 000002  | SANY = 000020     |
| C\$FLG 000015     | H\$LBDA 000070        | PSZ 000046R      | X\$SOTI= 000001  | \$BLANK= 000006   |
| C\$GNAM 000016    | H\$LBDN 000072        | PVCCTM 000004R   | X\$STIL= 000010  | \$CAT5 = ***** GX |
| C\$LCN 000010     | H\$LDTE 000002        | PVCDF 000000R    | X\$STIR= 000004  | \$CEACX= ***** GX |
| C\$LEN 000016     | H\$LEN 000042         | PVCFLG 000023R   | X\$SCHN= 000100  | \$DIGIT= 000024   |
| C\$LNK 000000     | H\$LOTS 000032        | PVCCLN 000002R   | X\$SDGR= 000002  | \$DNUMB= 000014   |
| C\$NAM 000002     | H\$NETW 000024        | PVCNAM 000024R   | X\$SFAS= 000200  | \$EOS = 000012    |
| C\$NML 000002     | H\$NML = 000006       | PVCOWN 000012R   | X\$SINC= 000004  | \$ERRNO 000040R   |
| C\$PORT 000014    | H\$NPT 000022         | PVCPRT 000022R   | X\$SIOC= 000010  | \$ERRN1 000100R   |
| C\$SCKP= 000000   | H\$PTB 000020         | PVCPSZ 000006R   | X\$SPVC= 000001  | \$ERRN2 000134R   |
| C\$SORE= 000400   | H\$PVC 000006         | PVCST 000000RG   | X\$SABQ 000076   | \$ERRN3 000166R   |
| C\$SRSH= 177564   | H\$RDTE 000004        | PVCT 000220R     | X\$ALF 000055    | \$ERR00= ***** GX |
| DLMCTL 000032R    | H\$RNW 000042         | PVCTL 001544R    | X\$AUC 000056    | \$ERR05= ***** GX |
| DLMIND 000035R    | H\$SVC 000036         | PVCWSZ 000010R   | X\$CLN= 000025   | \$ERR09= ***** GX |
| DLMPPV 000014R    | H\$TRB 000016         | PVDLM .J1446R    | X\$CTIM 000060   | \$ERRP8= ***** GX |
| DLM\$LT 000016R   | H\$XAVL 000100        | PVEND 000404R    | X\$DIAG 000003   | \$ERRIT= ***** GX |
| DLMSTA 000034R    | H\$XBIA 000074        | PVFG 000346R     | X\$DTE 000024    | \$EXIT = 000000   |
| DLMUNT 000033R    | H\$X29C 000040        | PVLCN 000162R    | X\$FLG 000014    | \$FAIL = 177777   |
| DTEDES= ***** GX  | INIXCB 001132R        | PVNAM 000056R    | X\$GLEN 000106   | \$HEADR= ***** GX |
| D\$BUG= 177514    | INVCIR 001536R        | PVNAMX= 000006   | X\$LCN 000026    | \$LAMD= 000000    |
| D\$ISK= 000000    | I\$RAR= 000000        | PVOWN 000310R    | X\$LEN 000106    | \$NUMBR= 000002   |
| D\$SL11= 000001   | I\$RDN= 000000        | PVPSZ 000244R    | X\$MOWN 000023   | \$OPVC 000000RG   |
| D\$SYNC= 000000   | K\$AR5 = ***** GX     | PVSTA 001554R    | X\$NLRE 000052   | \$RAD50= 000016   |
| D\$SYNM= 000000   | K\$CNT= 177546        | PVWSZ 000266R    | X\$NNRE 000054   | \$STRNG= 000004   |
| END 000156R       | K\$CSR= 177546        | P\$P45= 000000   | X\$NPL 000006    | \$SUBXP= 000010   |
| E\$XPR= 000000    | K\$LDL= 000000        | P\$SWRD= 000000  | X\$NPR 000007    | \$XALOC= ***** GX |
| FLAG 000020R      | K\$STPS= 000074       | Q\$SOPT= 000010  | X\$NRBY 000032   | \$SFLG= 177777    |
| FMT10 = ***** GX  | LD\$LP = 000000       | RTSPC = ***** GX | X\$NRPK 000042   | \$SKEY= 000002    |
| FMT8 = ***** GX   | LNKEND= ***** GX      | R\$R0 = 000002   | X\$NRRE 000053   | \$SSR = 000010    |
| FM.10 = 000000    | L\$ASG= 000000        | R\$R1 = 000004   | X\$NTBY 000036   | \$SSSTA= 000000   |
| FM.8 = 000000     | L\$DRV= 000000        | R\$R2 = 000006   | X\$NTPK 000046   | \$SSTMP= 000013R  |
| FNDHSH 001316R    | L\$P11= 000001        | R\$R3 = 000010   | X\$PKSZ 000016   | .PNUMB= ***** GX  |
| FNDPDV= ***** GX  | L\$S11R= 000000       | R\$R4 = 000012   | X\$PR 000011     | .PNUMH= ***** GX  |
| FNDPRT 001044R    | M\$SCRB= 000124       | R\$R5 = 000014   | X\$PRT 000021    | .PSTCN= ***** GX  |
| F\$LVL= 000001    | M\$SCRX= 000000       | R\$SDER= 000000  | X\$PS 000010     | .PSTPT= ***** GX  |
| G\$STP= 000000    | M\$SFCS= 000000       | R\$SK11= 000001  | X\$RPR 000013    | .TPARS= ***** GX  |
| G\$STSS= 000000   | M\$SMGE= 000000       |                  |                  |                   |
| .ABS. 000106      | 000 (RW,1,GBL,ABS,OV) |                  |                  |                   |

```

112 .SBTTL LOOK FOR RDT$DF MACRO
113
114 ;+
115 ;$QRDT - LOOK FOR RDT$DF MACRO
116
117 ; INPUTS:
118 ; NONE
119
120 ; OUTPUTS:
121 ; C-BIT=SUCCESS/FAILURE
122 ; R3,R4,R5=DESTROYED
123
124 000000 012705 000000' $QRDT:: MOV #RDTDF,R5 ; STATE TABLE ADDRESS
125 000004 005001 CLR R1 ; FULL KEYWORD MATCH LENGTH
126 000006 012702 000000' MOV #RDTKW,R2 ; KEYWORD TABLE ADDRESS
127 000012 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
128 000016 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
129 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
130 000026 CALL TPARS ; GO DO THE PARSE
131 000032 103003 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
132 000034 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
133 000040 001401 BEQ 101$ ; IF EQ, YES
134 000042 20$: RETURN
135
136 ;
137 ; ERROR CONDITION
138
139 000044 101$: MSG$R 1T ; SYNTAX ERROR

```

```

63                                     .SBTTL  LOCAL DATA STORAGE AND ERROR MESSAGES
64
65 000000                             .PSECT  DATA,D
66
67                                     ;
68                                     ; ERROR MESSAGES
69                                     ;
70
71                                     .ENABL  LC
72
73 000000                             NTLER$ ,9A,10,CERR,RTSPC,CFLIN,<occurrence of X29$DF>
74 000036                             NTLER$ ,9B,10,CERR,RTSPC,CFLIN,<value for max circuits>
75 000076                             NTLER$ ,9C,10,CERR,RTSPC,CFLIN,<counter timer value>
76 000132                             NTLER$ ,9D,8,CERR,RTSPC,CFLIN,<X.29 data block allocation failure>
77
78                                     .DSABL  LC
79
80
81                                     ;
82                                     ; LOCAL STORAGE FOR X29$DF PARAMETERS
83                                     ;
84 000206                             MAXVC: .BLKW 1                      ; MAXIMUM CIRCUITS
85 000210                             CTIMR: .BLKW 1                      ; COUNTER TIMER VALUE
86
87 000000                             .PSECT

```



```

84          .SBTTL  LOCAL DATA
85
86 000000    .PSECT  DATA,D
87
88          ; LOCAL DATA FOR CHN$DF MACRO
89          ;
90          CHNLLO: .BLKW  1          ; LOW END OF CHANNEL RANGE
91 000000    CHNLHI: .BLKW  1          ; HIGH END OF CHANNEL RANGE
92 000002
93
94          ; ERROR MESSAGES
95          ;
96          .ENABL  LC
97
98          NTLER$ ,08,8,CERR,RTSPC,CFLIN,<Channel block allocation failure>
99 000004
100          .DSABL  LC
101
102          .PSECT
103 000000
104
105

```

```

53      ;***
54      ; LIBRARY MACROS
55      ;***
56      .MCALL HWDDF$,LBDLF$,SWSTK$
57
58      ;***
59      ; LIBRARY SYMBOLS
60      ;***
61      000000      HWDDF$      ; FEATURES MASK BIT DEFINITIONS
62      000000      LBDLF$      ; TASK IMAGE LABFL BLOCK OFFSETS
63
64      ;***
65      ; LOCAL MACROS
66      ;***
67
68      ;
69      ; SHIFT LEFT
70      ;
71      .MACRO ASL$ COUNT,REG
72      .IF DF R$SEIS
73      ASH #'COUNT',REG
74      .IFF
75      .REPT COUNT
76      ASL REG
77      .ENDR
78      .ENDC
79      .ENDM ASL$
80
81      ;***
82      ; LOCAL SYMBOLS
83      ;***
84
85      ;
86      ; ERROR CODES RETURNED TO CALLER
87      ;
88      000000      .ASECT
89      000002      .=2      ; 0=SUCCESS
90      000002      ERR2: .BLKW 1      ; OPEN FAILURE
91      000004      ERR4: .BLKW 1      ; GET ATTRIBUTES FAILED
92      000006      ERR6: .BLKW 1      ; FILE NOT CONTIGUOUS
93      000010      ERR8: .BLKW 1      ; LABEL BLOCK READ FAILURE
94      000012      ERR10: .BLKW 1      ; INVALID PROCESS IMAGE
95      000014      ERR12: .BLKW 1      ; PROCESS NOT BUILT FOR APR5 (MAPPED)
96      000016      ERR14: .BLKW 1      ; PROCESS > 4K WORDS (MAPPED)
97      000020      ERR16: .BLKW 1      ; PROCESS EXTENSION > 4K WORDS (MAPPED)
98      000022      ERR18: .BLKW 1      ; PARTITION NOT IN SYSTEM
99      000024      ERR20: .BLKW 1      ; PROCESS/PARTITION BASE MISMATCH
100     000026      ERR22: .BLKW 1      ; PARTITION TOO SMALL
101     000030      ERR24: .BLKW 1      ; NOT COMMON PARTITION
102     000032      ERR26: .BLKW 1      ; PARTITION BUSY
103     000034      ERR28: .BLKW 1      ; CEX PARTITION NOT IN EXEC SPACE (MAPPED)
104     000036      ERR30: .BLKW 1      ; SUB-PCB ALLOCATION FAILURE (MAPPED)
105     000040      ERR32: .BLKW 1      ; MAIN PARTITION TOO FRAGMENTED (MAPPED)
106     000042      ERR34: .BLKW 1      ; PROCESS READ FAILURE
107     000044      ERR36: .BLKW 1      ; CETAB ALLOCATION FAILURE
108     000000      .PSECT

```

ELTIME - CALCULATE ELAPSED TIME MACRO V05.03b Saturday 29-Jun-85 17:42 Page 5-1  
 Symbol table

|                   |                  |                  |                  |                  |
|-------------------|------------------|------------------|------------------|------------------|
| A\$\$CHK= 000000  | ELTIME 000000RG  | K\$\$TPS= 000074 | N\$\$ACC= 000001 | R\$\$DER= 000000 |
| A\$\$CPS= 000000  | E\$\$XPR= 000000 | LD\$LP = 000000  | N\$\$BUF= 000001 | R\$\$K11= 000001 |
| A\$\$PRI= 000000  | F\$\$LVL= 000001 | L\$\$ASG= 000000 | N\$\$LDV= 000001 | R\$\$SND= 000000 |
| A\$\$TRP= 000000  | G\$\$TTP= 000000 | L\$\$DRV= 000000 | N\$\$MCP= 000001 | R\$\$11M= 000000 |
| C\$\$CKP= 000000  | G\$\$TSS= 000000 | L\$\$P11= 000001 | N\$\$MLL= 000001 | S\$\$WRG= 000000 |
| C\$\$SORE= 000400 | G\$\$ITK= 000000 | L\$\$11R= 000000 | N\$\$MOV= 000010 | S\$\$YSZ= 007600 |
| C\$\$RSH= 177564  | G\$\$WRD= 000000 | M\$\$CRB= 000124 | N\$\$NCT= 000001 | T\$\$KMG= 000000 |
| D\$\$BUG= 177514  | I\$\$RAR= 000000 | M\$\$CRX= 000000 | N\$\$PEM= 000001 | T\$\$MIN= 000000 |
| D\$\$ISK= 000000  | I\$\$RDN= 000000 | M\$\$FCS= 000000 | P\$\$P45= 000000 | V\$\$CTR= 001000 |
| D\$\$L11= 000001  | K\$\$CNT= 177546 | M\$\$MGE= 000000 | P\$\$WRD= 000000 | X\$\$DBT= 000000 |
| D\$\$SYNC= 000000 | K\$\$CSR= 177546 | M\$\$NET= 000000 | Q\$\$OPT= 000010 | ZTIM2 = ***** GX |
| D\$\$YNM= 000000  | K\$\$LDC= 000000 | M\$\$OVR= 000000 |                  |                  |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
 000022 001 (RW,I,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 8920 Words ( 35 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:05.14  
 SY:ELTIME.V2,[135,134]ELTIME/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLI9/ML,[130,10]RSXMCM/PA:1,[135,10]ELTIME

LBNIO      CREATED BY    MACRO    ON 29-JUN-85 AT 00:21      PAGE 1      E 16

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES  |
|---------|-------------|---|
| IO.RLB  | = ***** GX  | 6-82  |
| IS\$AS  | = *****     | 8-137      9-193                                      |
| LBNDPB  | = 000014 R  | #6-81      10-222                                     |
| L\$BHRB | = ***** GX  | 8-146   |
| M\$SMGE | = 000000    | 9-169   |
| PR7     | = ***** GX  | 9-172   |
| R\$S11D | = *****     | 8-137      9-193                                      |
| R\$S11M | = 000000    | 9-169   |
| STAT    | = 000002 R  | #6-76      7-113      7-115                           |
| UISAV   | = 000000 R  | #6-71      *9-175      9-186                          |
| USARO   | = ***** GX  | 9-173   |
| \$CLOPE | = 000000 RG | #7-112  |
| \$CLQST | = ***** GX  | 7-114   |
| \$IOSB  | = 000044 RG | 6-85      #6-96      10-224                           |
| \$LBN   | = 000036 RG | #6-90      *7-116      *7-117      *8-146      *8-147 |
| \$LBUF  | = 000030 RG | #6-87      *10-221                                    |
| \$LLEN  | = 000032 RG | #6-88      *8-133                                     |
| \$READ  | = 000152 RG | 8-134      9-182      #10-220                         |
| \$READX | = 000060 RG | #9-167  |
| \$RLBL  | = 000026 RG | #8-132  |

74  
75  
76  
77  
78  
79

.SBTTL LOCAL SYMBOL DEFINITIONS

LOCAL SYMBOL DEFINITIONS

000040

SPACE = 40

; ASCII SPACE

## MACRO CROSS REFERENCE

CREF 04.00

## MACRO NAME REFERENCES

| MACRO NAME | REFERENCES      | 12-212  | 12-220  | 12-254  | 13-267  | 13-273  | 13-275  | 13-282  | 13-306  | 14-338  |
|------------|-----------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| CALL       | 9-129<br>14-343 |         |         |         |         |         |         |         |         |         |
| CUGDFS     | #5-57           | 5-59    |         |         |         |         |         |         |         |         |
| DBGTPS     | #10-142         | #10-160 | #10-164 | #10-173 |         |         |         |         |         |         |
| EMSGSR     | #5-57           | 9-138   | 11-200  | 12-225  | 13-315  | 13-316  | 13-317  |         |         |         |
| ISTATS     | #5-57           | 10-142  |         |         |         |         |         |         |         |         |
| MTRANS     | #10-142         |         |         |         |         |         |         |         |         |         |
| NTLRS      | #5-57           | 8-101   | 8-102   | 8-103   | 8-104   |         |         |         |         |         |
| PHQDF      | #5-57           | 5-60    |         |         |         |         |         |         |         |         |
| RESMAP     | #5-69           | 13-307  | 14-356  |         |         |         |         |         |         |         |
| RETC       | #5-57           | 13-277  | 14-355  |         |         |         |         |         |         |         |
| RETURN     | 9-133           | 11-196  | 12-213  | 12-221  | 12-255  | 13-308  | 13-311  | 14-357  |         |         |
| SAVMAP     | #5-65           | 13-279  | 14-339  |         |         |         |         |         |         |         |
| STATS      | #5-57           | 10-146  | #10-149 | #10-152 | #10-155 | #10-158 | #10-162 | #10-166 | #10-171 | #10-175 |
| SWSTKS     | 13-273          | 14-338  |         |         |         |         |         |         |         |         |
| TRANS      | #5-57           | #10-147 | #10-150 | #10-153 | #10-156 | #10-159 | #10-160 | #10-163 | #10-164 | #10-167 |
|            | #10-172         | #10-173 |         |         |         |         |         |         |         |         |

```

286                                     .SBTTL DSC$DF ACTION ROUTINES
287
288
289                                     ; STORE CALL DATA MASK
290
291 000346 012700 000000G      DCM$K: MOV      #DSCMSK,R0      ; POINT TO CALL DATA MASK STORAGE
292 000352 012702 000000G      MOV      #DSCMCT,R2      ; POINT TO CALL DATA MASK COUNT
293 000356      CALL      STRHEX      ; STORE THE HEX VALUE
294 000362      BCS      101$      ; BR IF ILLEGAL VALUE
295 000364      RETURN
296
297                                     ; ERRORS
298
299 000366      101$: MSG$R M3      ; ILLEGAL CALL DATA MASK
300
301                                     ; STORE CALL DATA VALUE
302
303
304 000374 012700 000000G      DCVAL: MOV      #DC$VAL,R0      ; POINT TO CALL DATA VALUE STORAGE
305 000400 012702 000000G      MOV      #DSCVCT,R2      ; POINT TO CALL DATA VALUE COUNT
306 000404      CALL      STRHEX      ; STORE THE HEX VALUE
307 000410      BCS      101$      ; BR IF ILLEGAL VALUE
308 000412      RETURN
309
310                                     ; ERRORS
311
312 000414      101$: MSG$R M2      ; ILLEGAL CALL DATA VALUE
313
314 000422      STRHEX: SAVRG <R3>      ; SAVE R3
315 000424 016701 000000G      MOV      .PSTCN,R1      ; GET NUMBER OF CHARACTERS
316 000430 022701 000040      CMP      #DSCMX,R1      ; TOO MANY CHARACTERS?
317 000434 103436      BLO      40$      ; BR IF YES
318 000436 110112      MOV      R1,(R2)      ; STORE HEX DIGIT COUNT
319 000440 016702 000000G      MOV      .PSTPT,R2      ; GET ADDRESS OF SPECIFIED STRING
320 000444 012746 000001      MOV      #1-(SP)      ; INIT FLAG (0-LOW BITS,1-HIGH BITS)
321 000450 112203      10$: MOV      (R2)+,R3      ; GET CHARACTER
322 000452 122703 000106      CMP      #'F,R3      ; IS IT A LEGAL HEX VALUE?
323 000456 103425      BLO      40$      ; BR IF NO
324 000460 122703 000101      CMP      #'A,R3      ; IS IT A NUMBER?
325 000464 101003      BHI      15$      ; BR IF YES
326 000466 162703 000067      SUB      #67,R3      ; CONVERT ASCII HEX ALPHA TO BINARY
327 000472 000402      BR      17$      ; CONTINUE
328 000474 162703 000060      15$: SUB      #60,R3      ; CONVERT ASCII NUMBER TO BINARY
329 000500 005716      17$: TST      (SP)      ; STORE IN LOW BITS?
330 000502 001003      BNE      20$      ; BR IF NO
331 000504 150320      BIS      R3,(R0)+      ; ELSE STORE IN LOW FOUR BITS OF BYTE
332 000506 005216      INC      (SP)      ; SET FLAG TO STORE IN HIGH BITS
333 000510 000406      BR      30$      ; CONTINUE
334 000512      20$: ASL$      4,R3      ; MOVE TO HIGH FOUR BITS
335 000522 110310      MOV      R3,(R0)      ; STORE HEX VALUE
336 000524 005316      DEC      (SP)      ; SET FLAG TO STORE IN LOW BITS
337 000526 005301      30$: DEC      R1      ; MORE TO GET?
338 000530 003347      BGT      10$      ; BR IF YES
339 000532 005226      40$: INC      (SP)+      ; CLEAN UP STACK
340 000534      RESRG <R3>      ; RESTORE R3
341 000536      RETURN
342

```

```

147 .SBTTL TPARS STATE TABLES
148
149 ; TPARS STATE TABLES
150 ;
151 000372 ISTAT$ DSTST,DSTKW
152
153 ; DST$DF STATE TABLES
154 ;
155
156 000372 STATES$ DSTDF
157 000372 TRANS$ %DST$DF%,...1,SYNERR
158
159 000372 STATES$
160 000372 TRANS$ 'X25',,DSNM,,DS.X25,DSFLG ; TYPE
161 000372 TRANS$ 'X29',,DS.X29,DSFLG
162
163 000372 STATES$ DSNM
164 000372 TRANS$ '<','>'
165
166 000372 STATES$
167 000372 TRANS$ $STRNG,,DSNAM ; DESTINATION NAME
168
169 000372 STATES$
170 000372 TRANS$ '<','>'
171
172 000372 STATES$
173 000372 TRANS$ $NUMBR,,DSPRI ; PRIORITY
174
175 000372 STATES$
176 000372 TRANS$ '<','>'
177
178 000372 STATES$
179 000372 TRANS$ $NUMBR,,DSOBJ ; OBJECT NUMBER
180
181 000372 STATES$
182 000372 TRANS$ !END,$EXIT
183 000372 TRANS$ '<','>'
184
185 000372 STATES$
186 000372 TRANS$ $RAD50,,DSTSK ; TASK NAME
187
188 000372 STATES$ END
189 000372 TRANS$ $EOS,$EXIT
190 000372 TRANS$ '<';>,$EXIT
191
192 000372 STATES$

```



## MACRO CROSS REFERENCE

CREF 04.00

| MACRO NAME | REFERENCES      |                  |        |        |        |        |        |        |        |        |
|------------|-----------------|------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| CALL       | 8-136<br>16-442 | 12-267<br>16-449 | 13-291 | 13-303 | 14-327 | 14-332 | 15-360 | 15-361 | 15-368 | 15-413 |
| DBGTP\$    | #9-151          | #9-161           | #9-183 | #9-190 |        |        |        |        |        |        |
| DSTDF\$    | #5-60           | 5-63             |        |        |        |        |        |        |        |        |
| EMSG\$R    | #5-60           | 8-145            | 10-229 | 11-253 | 11-254 | 12-276 | 13-309 | 13-310 | 13-311 |        |
| ISTAT\$    | #5-61           | 9-151            |        |        |        |        |        |        |        |        |
| MTRAN\$    | #9-151          |                  |        |        |        |        |        |        |        |        |
| NILR\$     | #5-60           | 7-104            | 7-105  | 7-106  | 7-107  | 7-108  | 7-109  | 7-110  |        |        |
| PHBDF\$    | #5-60           | 5-64             |        |        |        |        |        |        |        |        |
| RESMAP     | #5-72           | 14-343           | 15-414 |        |        |        |        |        |        |        |
| RESRG      | #5-60           | 10-208           | 12-271 |        |        |        |        |        |        |        |
| RETC       | #5-60           | 14-342           | 15-363 |        |        |        |        |        |        |        |
| RETURN     | 8-140           | 10-225           | 11-247 | 12-272 | 13-305 | 14-344 | 15-415 | 16-450 |        |        |
| SAVMAP     | #5-68           | 14-328           | 15-365 |        |        |        |        |        |        |        |
| SAVRG      | #5-60           | 10-203           | 12-263 |        |        |        |        |        |        |        |
| STATE\$    | #5-61           | 9-156            | #9-159 | #9-163 | #9-166 | #9-169 | #9-172 | #9-175 | #9-178 | #9-181 |
|            | #9-185          | #9-188           | #9-192 |        |        |        |        |        |        |        |
| SWSTK\$    | 14-327          | 15-360           |        |        |        |        |        |        |        |        |
| TRANS      | #5-61           | #9-157           | #9-160 | #9-161 | #9-164 | #9-167 | #9-170 | #9-173 | #9-176 | #9-179 |
|            | #9-182          | #9-183           | #9-186 | #9-189 | #9-190 |        |        |        |        |        |

```

317
318
319      ; NETWORK NAME
320
321 000310 016700 000000G DTENET: MOV      .PSTCN,R0      ; GET SIZE OF SUPPLIED NAME
322 000314 020027 000006      CMP      R0,#DTNMMX      ; BIGGER THAN MAXIMUM?
323 000320 101010      BHI      101$      ; BR IF YES, ERROR
324 000322 016701 000000G      MOV      .PSTPT,R1      ; POINT TO NAME
325 000326 012702 000027'      MOV      #DTNTNM,R2      ; POINT TO LOCAL COPY
326 000332 112122      10$: MOVB      (R1)+,(R2)+      ; COPY NAME
327 000334 005300      DEC      R0      ; MORE TO COPY?
328 000336 003375      BGT      10$      ; BR IF YES
329 000340 000403      BR      DTEEND      ; ELSE MERGE INTO END PROCESSING
330
331      ; ERRORS
332
333
334 000342      101$: MSG$R 00      ; ILLEGAL NETWORK NAME PARAMETER
335

```

\*\*FILE\*\*ID\*\*CFPPSN

F 7

```
CCCCCCCC FFFFFFFF PPPPPPP PPPPPPP SSSSSSS NN NN
CCCCCCCC FFFFFFFF PPPPPPP PPPPPPP SSSSSSS NN NN
CC FF PP PP PP SS NN NN
CC FF PP PP PP SS NN NN
CC FF PP PP PP SS NNNN NN
CC FFFFFFFF PPPPPPP PPPPPPP SSSSSS NN NN NN
CC FFFFFFFF PPPPPPP PPPPPPP SSSSSS NN NN NN
CC FF PP PP PP SS NN NNNN
CC FF PP PP PP SS NN NNNN
CC FF PP PP PP SS NN NN
CC FF PP PP PP SS NN NN
CCCCCCCC FF PP PP SSSSSSS NN NN
CCCCCCCC FF PP PP SSSSSSS NN NN
```

```
LL SSSSSSS TTTTTTTTT
LL SSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSS TT
LL SSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLL SSSSSSS TT
LLLLLLLLL SSSSSSS TT
```

```

356 000572 016760 000012' 000034      MOV     PSNHI,H$HITS(R0)      ; AND HIGH TRANSPORT SUBADDRESS
357 000600                                60$:     RETURN
358                                         ;
359                                         ; ERROR MESSAGES
360                                         ;
361 000602                                101$:    MSG$R  D2          ; ERROR - INVALID HIGH SUBADDRESS VALUE
362 000610                                102$:    MSG$R  P5          ; ERROR - HOME BLOCK ALLOCATION FAILURE
363 000616                                103$:    MSG$R  P6          ; ERROR - PORT TABLE ALLOCATION FAILURE
364                                         .DSABL  LSB
365                                000001      .END

```

```

196          .SBTTL  TPARS STATE TABLES
197
198          ; TPARS STATE TABLES
199          ;
200          ISTAT$  PVCST,PCVKW
201
202          ; PVC$DF STATE TABLES
203          ;
204          STATES$ PVCDF
205          TRANS$  %PVC$DF%,,,1,SYNERR
206
207          STATES$
208          TRANS$  $RAD50,,PVNAM          ; CHANNEL IDENTIFICATION
209
210          STATES$
211          TRANS$  !CHKDLM,$EXIT          ; IGNORE IF DLM CIRCUIT
212          TRANS$  <','>
213
214          STATES$
215          TRANS$  CHNLNO
216          TRANS$  $NUMBR,,PVLN          ; LOGICAL CHANNEL NUMBER
217
218          STATES$
219          TRANS$  <','>
220
221          STATES$
222          TRANS$  $NUMBR,,PVCT          ; COUNTER TIMER
223
224          STATES$
225          TRANS$  <','>
226
227          STATES$
228          TRANS$  'OFF',,PSZ          ; STATE
229          TRANS$  'ON',,PSZ
230
231          STATES$ PSZ
232          TRANS$  !END,$EXIT,PVEND
233          TRANS$  <','>
234
235          STATES$
236          TRANS$  $NUMBR,WND$SZ,PVPSZ    ; PACKET SIZE
237          TRANS$  $LAMD$
238
239          STATES$ WND$SZ
240          TRANS$  !END,$EXIT,PVEND
241          TRANS$  <','>
242
243          STATES$
244          TRANS$  $NUMBR,OWNER,PVWSZ    ; WINDOW SIZE
245          TRANS$  $LAMD$
246
247          STATES$ OWNER
248          TRANS$  !END,$EXIT,PVEND
249          TRANS$  <','>
250
251          STATES$
252          TRANS$  $RAD50,PFL$GS,PVOWN    ; PROCESS OWNER

```

CFPPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:18 Page 20-2  
Symbol table

|         |        |     |                    |
|---------|--------|-----|--------------------|
| DATA    | 001600 | 001 | (RW,I,LCL,REL,CON) |
|         | 000232 | 002 | (RW,D,LCL,REL,CON) |
| \$STATE | 000214 | 003 | (RW,D,LCL,REL,CON) |
| \$KTAB  | 000006 | 004 | (RW,D,LCL,REL,CON) |
| \$KSTR  | 000016 | 005 | (RW,D,LCL,REL,CON) |

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 15212 Words ( 60 Pages)  
Size of core pool: 16552 Words ( 63 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:55.09

SY:CFPPVC.V2,[132,134]CFPPVC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFPPVC

```

141          .SBTTL  TPARS STATE TABLES
142          ;
143          ; TPARS STATE TABLES
144          ;
145          ISTAT$  RDTST,RTKW
146          ;
147          ; RDT$DF
148          ;
149          STATES$  RDTDF
150          TRANS$   %RDT$DF%,.,1,SYNERR
151          ;
152          STATES$
153          TRANS$   $STRNG,,RDNAM          ; DTE NAME
154          ;
155          STATES$
156          TRANS$   <','>
157          ;
158          STATES$
159          TRANS$   $DIGIT,,RDTEST          ; START OF DTE ADDRESS
160          ;
161          STATES$  RDTENX
162          TRANS$   $DIGIT,RDTENX,RTENO
163          TRANS$   !END,$EXIT,RDEND          ; END OF DTE ADDRESS
164          ;
165          ; CHECK FOR END OF SOURCE LINE
166          ;
167          STATES$  END
168          TRANS$   <','>,$EXIT
169          TRANS$   $EOS,$EXIT
170          ;
171          STATES$

```

```

89          .SBTTL LOOK FOR X29$DF MACRO
90
91          ;+
92          ; $QX29 - LOOK FOR X29$DF MACRO
93          ;
94          ; INPUTS:
95          ;     NONE
96          ;
97          ; OUTPUTS:
98          ;     C-BIT=SUCCESS/FAILURE
99          ;     R3,R4,R5=DESTROYED
100         ; -
101
102         000000 012705 000000'   $QX29:: MOV    #X29DF,R5   ; STATE TABLE ADDRESS
103         000004 005001           CLR    R1               ; FULL KEYWORD MATCH LENGTH
104         000006 012702 000000'   MOV    #X29KW,R2       ; KEYWORD TABLE ADDRESS
105         000012 016703 000000G   MOV    CFGSZ,R3        ; RECORD LENGTH
106         000016 012704 000000G   MOV    #CFGBF,R4       ; RECORD BUFFER ADDRESS
107         000022 005067 000000G   CLR    SYNERR         ; CLEAR SYNTAX ERROR FLAG
108         000026           CALL    TPARS                 ; GO DO THE PARSE
109         000032 103003           BCC    20$              ; IF CC, FOUND WHAT WE WERE LOOKING FOR
110         000034 005367 000000G   DEC    SYNERR          ; DID SYNTAX ERROR OCCUR?
111         000040 001401           BEQ    101$             ; IF EQ, YES
112         000042           20$: RETURN
113
114          ;
115          ; ERROR CONDITION
116
117         000044           101$: MSG$R 1T                ; SYNTAX ERROR

```



```

107          .SBTTL LOOK FOR ***$DF MACRO
108
109          ;
110          ; Q*** - LOOK FOR ***$DF MACRO
111          ;
112          ; INPUTS:
113          ;     NONE
114          ;
115          ; OUTPUTS:
116          ;     C-BIT=SUCCESS/FAILURE
117          ;     R3,R4,R5=DESTROYED
118          ;
119          ; ENABL LSB
120
121 000000 012705 000000' $QX3P:: MOV #X3PDF,R5 ; STATE TABLE ADDRESS
122 000004 000402          BR 10$
123 000006 012705 000036' $QCHN:: MOV #CHNDF,R5 ; STATE TABLE ADDRESS
124
125 000012 005001          10$: CLR R1 ; FULL KEYWORD MATCH LENGTH
126 000014 012702 000000' MOV #X3PKW,R2 ; KEYWORD TABLE ADDRESS
127 000020 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
128 000024 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
129 000030 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
130 000034          CALL TPARS ; GO DO THE PARSE
131 000040 103003          BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
132 000042 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
133 000046 001401          BEQ 101$ ; IF EQ, YES
134 000050          20$: RETURN
135
136          ;
137          ; ERROR CONDITION
138          ;
139 000052          101$: MSG$R 1T ; SYNTAX ERROR
140
141          .DSABL LSB

```

```
110          ;***
111          ; LOCAL DATA
112          ;***
113
114          .PSECT DATA,D
115
116          ;
117          ; ERROR INDICATION
118
119          ERROR: .BLKW 1
120
121          .PSECT ..3UF,D
122
123          ;
124          ; DISK BUFFER FOR LABEL BLOCK READ
125          ;
126          DSKBUF: .BLKW 1
127
128          .PSECT
```

ELTIME      CREATED BY MACRO ON 29-JUN-85 AT 17:42      PAGE 1      F 15  
SYMBOL CROSS REFERENCE      CREF      04.00  
SYMBOL      VALUE      REFERENCES  
ELTIME      000000 RG      #5-73  
ZTIME      =      \*\*\*\*\* GX      5-74

LBNIO      CREATED BY    MACRO    ON 29-JUN-85 AT 00:21      PAGE 2      F 16

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES   |
|------------|--|
| CALL       | 7-114      8-134      9-171      9-180      9-183  |
| DIR\$      | #5-59      10-222                                  |
| QIOW\$     | #5-59  |
| RETURN     | 7-118      8-149      9-177      9-189      10-227 |
| SWSTK\$    | 9-171      9-183                                   |

```

81 LOCAL DATA
82
83 .SBTTL LOCAL DATA
84
85 ; LOCAL DATA FOR CUG$DF MACRO
86 ;
87 .PSECT DATA,D
88 .NLIST BEX
89
90 CUGNAM: .BLKB 6 ; CLOSED USER GROUP NAME
91 000000 CUGNLN = 4 ; MAXIMUM DIGITS IN CUG NUMBER
92 000004 CUGNUM: .BLKB CUGNLN ; CLOSED USER GROUP NUMBER
93 000011' CUGEND=-1 ; END OF CUG NAME
94 000012 CUGPCK: .BLKB CUGNLN/2 ; CUG NUMBER IN BCD FORMAT
95 000014 CUGFLG: .BLKB 1 ; FLAGS BYTE
          .EVEN

```

\*\*FILE\*\*ID\*\*CFPDSA

|          |          |          |          |          |            |      |
|----------|----------|----------|----------|----------|------------|------|
| CCCCCCCC | FFFFFFFF | PPPPPPPP | DDDDDDDD | SSSSSSSS | AAAAAA     |      |
| CCCCCCCC | FFFFFFFF | PPPPPPPP | DDDDDDDD | SSSSSSSS | AAAAAA     |      |
| CC       | FF       | PP PP    | DD DD    | SS       | AA AA      |      |
| CC       | FF       | PP PP    | DD DD    | SS       | AA AA      |      |
| CC       | FF       | PP PP    | DD DD    | SS       | AA AA      |      |
| CC       | FFFFFFFF | PPPPPPPP | DD DD    | SSSSSS   | AA AA      |      |
| CC       | FFFFFFFF | PPPPPPPP | DD DD    | SSSSSS   | AA AA      |      |
| CC       | FF       | PP       | DD DD    | SS       | AAAAAAAAAA |      |
| CC       | FF       | PP       | DD DD    | SS       | AAAAAAAAAA |      |
| CC       | FF       | PP       | DD DD    | SS       | AA AA      | .... |
| CC       | FF       | PP       | DD DD    | SS       | AA AA      | .... |
| CCCCCCCC | FF       | PP       | DDDDDDDD | SSSSSSSS | AA AA      | .... |
| CCCCCCCC | FF       | PP       | DDDDDDDD | SSSSSSSS | AA AA      | .... |

|            |          |          |
|------------|----------|----------|
| LL         | SSSSSSSS | TTTTTTTT |
| LL         | SSSSSSSS | TTTTTTTT |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SSSSSS   | TT       |
| LL         | SSSSSS   | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LLLLLLLLLL | SSSSSSSS | TT       |
| LLLLLLLLLL | SSSSSSSS | TT       |

343

000001

.END

```

194                                     .SBTTL  DST$DF ACTION ROUTINES
195
196                                     ;
197                                     ; DESTINATION NAME
198
199 000372 016702 000000G  DSNAM:  MOV    .PSTCN,R2      ; GET NUMBER OF CHARACTERS MATCHED
200 000376 022702 000006    CMP    #DSNAMX,R2      ; LEGAL DESTINATION NAME?
201 000402 103447          BLO    101$             ; BR IF NO
202 000404 012700 000000G    MOV    #DSNAM,R0      ; POINT TO STORAGE FOR DESTINATION NAME
203 000410          SAVRG  <R0>                   ; SAVE IT
204 000412 012701 000006    MOV    #DSNAMX,R1      ; GET LENGTH OF NAME
205 000416 112720 000040 10$:  MOVB  #SPACE,(R0)+  ; INITIALIZE DESTINATION NAME
206 000422 005301          DEC    R1              ; MORE TO INITIALIZE?
207 000424 003374          BGT    10$             ; BR IF YES
208 000426          RESRG  <R0>                   ; RECOVER DESTINATION NAME ADDRESS
209 000430 016701 000000G    MOV    .PSTPT,R1      ; GET ADDRESS OF SPECIFIED NAME
210 000434 112120          MOVB  (R1)+,(R0)+      ; STORE SPECIFIED NAME
211 000436 005302          DEC    R2              ; MORE TO STORE?
212 000440 003375          BGT    20$             ; BR IF YES
213 000442 012702 000000G    MOV    #CUNMMX,R2      ; GET LENGTH OF CUG NAME
214 000446 012701 000000G    MOV    #DSACUG,R1      ; GET ADDRESS OF CUG NAME STORAGE
215 000452 112721 000040 30$:  MOVB  #SPACE,(R1)+  ; INITIALIZE TO SPACES
216 000456 005302          DEC    R2              ; MORE TO INITIALIZE?
217 000460 003374          BGT    30$             ; BR IF YES
218 000462 005067 000000G    CLR    DSTTSK        ; ASSUME NO TASK NAME
219 000466 005067 000002G    CLR    DSTTSK+2      ;
220 000472 005067 000000G    CLR    DSASLO        ; DEFAULT LOW SUBADDRESS TO ZERO
221 000476 012767 177777 000000G  MOV    #-1,DSASHI  ; ASSUME NO SUBADDRESS RANGE
222 000504 105067 000000C    CLRB  DSCMCT        ; CLEAR CALL DATA MASK COUNT
223 000510 105067 000000G    CLRB  DSCVCT        ; CLEAR CALL DATA VALUE COUNT
224 000514 105067 000000G    CLRB  DSARCT        ; CLEAR REMOTE DTE ADDRESS COUNT
225 000520          RETURN
226
227                                     ; ERRORS
228
229 000522 101$:  MSG$R  N4                      ; ILLEGAL DESTINATION NAME

```



|          |          |         |         |          |          |      |
|----------|----------|---------|---------|----------|----------|------|
| CCCCCCCC | FFFFFFFF | PPPPPPP | DDDDDDD | TTTTTTTT | EEEEEEEE |      |
| CCCCCCCC | FFFFFFFF | PPPPPPP | DDDDDDD | TTTTTTTT | EEEEEEEE |      |
| CC       | FF       | PP PP   | DD DD   | TT       | EE       |      |
| CC       | FF       | PP PP   | DD DD   | TT       | EE       |      |
| CC       | FF       | PP PP   | DD DD   | TT       | EE       |      |
| CC       | FF       | PP PP   | DD DD   | TT       | EE       |      |
| CC       | FFFFFFFF | PPPPPPP | DD DD   | TT       | EEEEEEEE |      |
| CC       | FFFFFFFF | PPPPPPP | DD DD   | TT       | EEEEEEEE |      |
| CC       | FF       | PP      | DD DD   | TT       | EE       |      |
| CC       | FF       | PP      | DD DD   | TT       | EE       |      |
| CC       | FF       | PP      | DD DD   | TT       | EE       | .... |
| CC       | FF       | PP      | DD DD   | TT       | EE       | .... |
| CCCCCCCC | FF       | PP      | DDDDDDD | TT       | EEEEEEEE | .... |
| CCCCCCCC | FF       | PP      | DDDDDDD | TT       | EEEEEEEE | .... |

|           |         |          |
|-----------|---------|----------|
| LL        | SSSSSSS | TTTTTTTT |
| LL        | SSSSSSS | TTTTTTTT |
| LL        | SS      | TT       |
| LL        | SS      | TT       |
| LL        | SS      | TT       |
| LL        | SS      | TT       |
| LL        | SSSSSS  | TT       |
| LL        | SSSSSS  | TT       |
| LL        | SS      | TT       |
| LL        | SS      | TT       |
| LL        | SS      | TT       |
| LL        | SS      | TT       |
| LLLLLLLLL | SSSSSSS | TT       |
| LLLLLLLLL | SSSSSSS | TT       |

```

337
338
339
340 000350
341 000354 103561
342 000356 020427 000020
343 000362 103167
344 000364
345 000370 103556
346
347 000372 016701 000000G
348 000376 006301
349 000400 006301
350 000402 062701 000122
351
352 000406
353 000412
354 000416 103006
355 000420
356 000432 000527
357
358 000434
359 000440 010067 000000G
360 000444 010046
361 000446
362 000452 012603
363 000454 010302
364 000456 105022
365 000460 005301
366 000462 003375
367
368 000464 010301
369 000466 062701 000020
370 000472 012702 000010
371 000476 012705 000017
372 000502 112521
373 000504 005302
374 000506 003375
375 000510 116763 000000G 000017
376
377 000516 010301
378 000520 062701 000114
379 000524 012702 000006
380 000530 012705 000027
381 000534 112521
382 000536 005302
383 000540 003375
384
385 000542 110463 000004
386 000546 006304
387 000550 017702 000000G
388
389 000554 060204
390 000556 111463 000016
391 000562 016763 000000G 000040
392 000570 016700 000000G
393 000574 062700 000002

; END OF DTE$DF MACRO
DTEEND: CALL CHKDTE ; CHECK FOR VALID DTE ADDRESS
; ERROR - DUPLICATE DTE ADDRESS
; MAXIMUM DTEs ALREADY DEFINED?
; YES- THIS ONE IS TOO MANY
; CHECK IF LINE EXISTS
; BR IF LINE NOT IN SYSTEM
BCS 101$
CMP R4,#NUMDTE
BHS 104$
CALL CHKDEV
BCS 102$
; GET NUMBER OF ENTRIES IN HASH
; TIMES 4 TO GET
; SIZE IN BYTES
; PLUS DTE SIZE GETS TOTAL SIZE
MOV HSHSZ,R1
ASL R1
ASL R1
ADD #L$LEN,R1
; ENTER SYSTEM STATE
; ALLOCATE BLOCK FROM EXTENDED POOL
; BR IF SUCCESSFUL ALLOCATION
; SET USER C-BIT
; AND EXIT
SWSTK$ 120$
CALL $XALOC
BCC 40$
RETC R0
BR 110$
40$: SAVMAP ; SAVE CURRENT MAPPING
MOV R0,DTEDES ; SAVE DTE DESCRIPTOR ADDRESS
MOV R0,-(SP) ; GET UNMAPPED ADDRESS
CALL $CEACK ; CONVERT TO MAPPED ADDRESS
MOV (SP)+,R3 ; SAVE MAPPED ADDRESS
MOV R3,R2 ; POINT TO DTE DESCRIPTOR BLOCK
42$: CLRB (R2)+ ; INITIALIZE BLOCK
DEC R1 ; MORE TO INITIALIZE?
BGT 42$ ; BR IF YES
; GET ADDRESS OF DTE DESCRIPTOR BLOCK
; POINT TO DTE ADDRESS FIELD
; GET LENGTH OF ADDRESS FIELD
; GET ADDRESS OF SPECIFIED ADDRESS
; STORE DTE ADDRESS
; MORE TO STORE?
; BR IF YES
; STORE DTE ADDRESS LENGTH
MOV R3,R1
ADD #L$DTEA,R1
MOV #<DTEALN+1>/2,R2
MOV #DTEPCK,R5
45$: MOVB (R5)+,(R1)+
DEC R2
BGT 45$
MOVB COUNT,L$DTEL(R3)
; GET ADDRESS OF BLOCK
; POINT TO NETWORK NAME FIELD
; GET LENGTH OF NAME
; POINT TO SPECIFIED NAME
; STORE NETWORK NAME
; MORE TO STORE?
; BR IF YES
50$: MOVB R3,R1
ADD #L$NETW,R1
MOV #DINMMX,R2
MOV #DINTNM,R5
MOVB (R5)+,(R1)+
DEC R2
BGT 50$
; STORE SYSTEM LINE NUMBER
; GET SYSTEM LINE NUMBER X 2
; POINT TO REVERSE MAPPING TABLE
MOVB R4,L$SLN(R3)
ASL R4
MOV #LLCTA,R2
; INDEX INTO REVERSE MAPPING TABLE
; STORE CHANNEL NUMBER
; STORE COUNTER TIMER
; GET TIMER ADDRESS
ADD R2,R4
MOVB (R4),L$LLCH(R3)
MOV (TIM,L$CTIM(R3))
MOV ZTIME,R0
ADD #2,R0

```

CFPPSN - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:17 <sup>6 7</sup>  
Table of contents

|     |     |                         |
|-----|-----|-------------------------|
| 5-  | 55  | MACRO DEFINITIONS       |
| 6-  | 80  | LOCAL SYMBOLS           |
| 7-  | 96  | LOCAL DATA              |
| 8-  | 113 | ERROR MESSAGES          |
| 9-  | 131 | LOOK FOR PSN\$DF MACRO  |
| 10- | 160 | TPARS STATE TABLES      |
| 11- | 210 | PSN\$DF ACTION ROUTINES |

```

Symbol table
ASSCHK= 000000      HSDST 000012      K$STPS= 000074      PSNLO 000010R      002 $CEACK= ***** GX
ASSCPS= 000000      HSD29 000014      LD$LP = 000000      PSNNAM 000000R      002 $DIGIT= 000024
ASSPRI= 000000      HSFLG 000000      L$SAG= 000000      PSNNM 000052R      $DNUMB= 000014
ASSTRP= 000000      H$GLEN 000104      L$SDRV= 000000      PSNSHI 000330R      $E0S = 000012
CERR = ***** GX  H$GLT 000044      L$SP11= 000001      PSNSLO 000242R      $ERRD1 000274R      002
CFGBF = ***** GX H$GNAM 000050      L$ST1R= 000000      PSNST 000000RG      003 $ERRD2 000332R      002
CFGSZ = ***** GX H$GNML= 000020      M$CRB= 000124      P$P45= 000000      $ERRP1 000014R      002
CFLIN = ***** GX H$GPT 000046      M$CRX= 000000      P$WRD= 000000      $ERRP2 000052R      002
CHKSUB 000276R      H$HITS 000034      M$FCS= 000000      Q$OPT= 000010      $ERRP3 000100R      002
C$CKP= 000000      H$HLEN 000044      M$MGE= 000000      RTSPC = ***** GX $ERRP4 000130R      002
C$DRE= 000400      H$LBDA 000070      M$NET= 000000      R$R0 = 000002      $ERRP5 000154R      002
C$RSH= 177564      H$LBDN 000072      M$OVR= 000000      R$R1 = 000004      $ERRP6 000226R      002
D$BUG= 177514      H$LDTE 000002      N$ACC= 000001      R$R2 = 000006      $ERR1T= ***** GX
D$LSK= 000000      H$LEN 000042      N$BUF= 000001      R$R3 = 000010      $EXIT = 000000
D$LL1= 000001      H$LOTS 000032      N$LDV= 000001      R$R4 = 000012      $FAIL = 177777
D$SYNC= 000000      H$NETW 000024      N$MCP= 000001      R$R5 = 000014      $HEADR= ***** GX
D$SYNM= 000000      H$NML = 000006      N$MML= 000001      R$SDER= 000000      $LAMDA= 000000
END 000104R      003 H$NPT 000022      N$MOV= 000010      R$SK11= 000001      $NUMBR= 000002
E$XPR= 000000      H$PTB 000020      N$NCT= 000001      R$SND= 000000      $QPSN 000000RG
FMT10 = ***** GX H$PVC 000006      N$PEM= 000001      R$S1M= 000000      $RAD50= 000016
FMT8 = ***** GX  H$RDTE 000004      PNAMX= 000006      R$S1M= 000000      $STRNG= 000004
FM.10 = 000000      H$RNW 000042      PORT 000160R      SUBAMX= ***** GX
FM.8 = 000000      H$SVC 000036      PORTMX= 000400      SYNERR= ***** GX
F$LVL= 000001      H$STRB 000016      PORTNO= ***** GX
G$TTP= 000000      H$XAVL 000100      PS$H 000076R      S$WRG= 000000
G$TSS= 000000      H$XBJA 000074      PSIPT = ***** GX  003 S$YSZ= 007600
G$TTK= 000000      H$X29C 000040      PSLO 000054R      T$KMG= 000000
G$WRD= 000000      I$RAR= 000000      PSND 000000R      003 T$MIN= 000000
HF$DLM= 000002      I$RDN= 000000      PSNEND 000356R      003 V$CTR= 001000
HF$GWY= 000010      K$CNT= 177546      PSNFG 000216R      X$DBT= 000000
HF$HDS= 000004      K$CSR= 177546      PSNFLG 000006R      $ALL16= ***** GX
HF$XDF= 000020      K$LDC= 000000      PSNHI 000012R      $ALPHA= 000022
H$CUG 000010      K$LDL= 000000      PSNKG 000000RG  002 $ANY = 000020
                                004 $BLANK= 000006
                                005

```

```

. ABS. 000104 000 (RW,I,GBL,ABS,OVR)
      000624 001 (RW,I,LCL,REL,CON)
DATA 000370 002 (RW,D,LCL,REL,CON)
$STATE 000114 003 (RW,D,LCL,REL,CON)
$KTAB 000002 004 (RW,D,LCL,REL,CON)
$KSTR 000007 005 (RW,D,LCL,REL,CON)
Errors detected: 0

```

\*\*\* Assembler statistics

```

Work file reads: 0
Work file writes: 0
Size of work file: 13027 Words ( 51 Pages)
Size of core pool: 14440 Words ( 55 Pages)
Operating system: RSX-11M/PLUS

```

```

Elapsed time: 00:00:33.15
SY:CFPPSN.V2,[132,134]CFPPSN/CR/-SP=SY:[1,1]RSXCMC.SML/ML,[130,110]NETLIB/ML,[130,10]RSXCMC/PA:1,[132,10]CFPPSN

```

```

253 000056          TRANS $LAMDA
254
255 000056          STATES$ PFLAGS
256 000056          TRANS $!END,$EXIT,PVEND
257 000056          TRANS $<'>
258
259 000056          STATES$
260 000056          TRANS $NUMBR,END,PVFG          ; FLAGS BYTE
261 000056          TRANS $LAMDA
262
263 000056          STATES$ END
264 000056          TRANS $EOS,$EXIT
265 000056          TRANS $<'>,$EXIT
266
267          ; CHECK FOR DLM CIRCUIT NAME - SUBEXPRESSION
268          ;
269 000056          STATES$ CHKDLM
270 000056          TRANS $'-,,PVDLM
271
272 000056          STATES$
273 000056          TRANS $DNUMB,,PVCTL
274
275 000056          STATES$
276 000056          TRANS $'
277
278 000056          STATES$
279 000056          TRANS $DNUMB,,PVSTA,PF.DLM,FLAG
280
281 000056          STATES$
282 000056          TRANS $<'>,$EXIT
283
284 000056          STATES$

```

CFPPVC CREATED BY MACRO ON 29-JUN-85 AT 00:19 PAGE 1 G 10  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE      | REFERENCES  |
|---------|------------|---|
| ALLXCB  | 000640 R   | 13-430 #14-461  |
| BIASX   | 001426 R   | 17-580 #19-654  |
| BLKSZD  | = ***** GX | 11-304  |
| CERR    | = ***** GX | 8-157 8-158 8-159                                       |
| CFGBF   | = ***** GX | 9-183   |
| CFGSZ   | = ***** GX | 9-182   |
| CFLIN   | = ***** GX | 8-156 8-157 8-158 8-159                                 |
| CHKBLK  | = ***** GX | 12-346  |
| CHKPVC  | 000742 R   | 13-401 #15-496  |
| CHKWND  | = ***** GX | 12-358  |
| CHNLMX  | = ***** GX | 11-321  |
| CSDTE   | 000012     | *13-424   |
| C\$FLG  | 000015     | *13-422   |
| C\$LCN  | 000010     | *13-423   |
| C\$LEN  | 000016     | 13-403  |
| C\$NAM  | 000002     | 13-416 15-509   |
| C\$PORT | 000014     | *13-429   |
| DLMCTL  | 000032 R   | #7-141 *20-700  |
| DLMIND  | 000035 R   | #7-144 *20-685  |
| DLMPDV  | 000014 R   | #7-134 *20-687  |
| DLMSLT  | 000016 R   | #7-135  |
| DLMSTA  | 000034 R   | #7-143 *20-702  |
| DLMUNT  | 000033 R   | #7-142  |
| DTEDES  | = ***** GX | 13-424 17-584 18-616                                    |
| FLAG    | 000020 R   | #7-136 *9-178   |
| FMT10   | = ***** GX | 8-156 8-157 8-158                                       |
| FMT8    | = ***** GX | 8-159   |
| FM.10   | = 000000   | #8-156 #8-157 #8-158                                    |
| FM.8    | = 000000   | #8-159  |
| FNDHSH  | 001316 R   | 14-470 #18-615  |
| FNDPDV  | = ***** GX | 14-474  |
| FNDPRT  | 001044 R   | 13-425 #16-535  |
| HSADD   | = ***** GX | 18-627  |
| HSBSZ   | = ***** GX | 18-621 18-631   |
| H\$PTB  | 000020     | 14-465 16-537   |
| H\$PVC  | 000006     | 13-434 15-497   |
| INIXCB  | 001132 R   | 14-469 #17-571  |
| INVCIR  | 001536 R   | 20-677 20-690 #20-695 20-704 20-706                     |
| KSAR5   | = ***** GX | 13-406 13-436 15-500 15-515 16-535 16-552 18-615 18-638 |
| LNKEND  | = ***** GX | 13-435  |
| N\$SVCT | = *****    | 13-409 13-427 13-432 15-514 20-683                      |
| PCVKW   | 000000 RG  | 9-181 #10-200   |
| PDVID   | = ***** GX | 20-681  |
| PDVTA   | = ***** GX | 20-686  |
| PF.DLM  | = 000001   | #7-137  |
| PORTNO  | = ***** GX | 16-549  |
| PRTADD  | 000000 R   | #7-128 14-468 *16-546                                   |
| PSIPT   | = ***** GX | 14-464 15-496 16-536                                    |
| PVCCTM  | 000004 R   | #7-130 *11-335 17-583                                   |
| PVCDF   | 000000 R   | 9-179   |
| PVCFLG  | 000023 R   | #7-139 *11-307 *12-389 13-422 18-620 18-636             |
| PVCLCN  | 000002 R   | #7-129 *11-323 13-423 17-586                            |

```

173                                     .SBTTL RDT$DF ACTION ROUTINES
174
175                                     ; REMOTE DTE NAME
176
177
178 000052 016700 000000G  RDNAM: MOV    .PSTCN,R0      ; GET NUMBER OF CHARACTERS SPECIFIED
179 000056 012702 000006  MOV    #RDTEMX,R2      ; GET MAXIMUM ALLOWED
180 000062 020002 000000  CMP     R0,R2          ; LEGAL NAME LENGTH?
181 000064 101016 000000'  BHI     101$      ; BR IF NO
182 000066 012701 000000'  MOV    #RDTNAM,R1      ; POINT TO REMOTE DTE NAME STORAGE
183 000072 112721 000040  5$:  MOVB   #SPACE,(R1)+    ; FILL WITH SPACES
184 000076 005302 000000'  DEC     R2          ; MORE TO PAD?
185 000100 003374 000000'  BGT     5$          ; BR IF YES
186 000102 012701 000000'  MOV    #RDTNAM,R1      ; POINT TO REMOTE DTE NAME STORAGE
187 000106 016702 000000G  MOV    .PSTPT,R2      ; POINT TO SPECIFIED NAME
188 000112 112221 000000'  10$: MOVB   (R2)+,(R1)+    ; STORE SPECIFIED REMOTE DTE NAME
189 000114 005300 000000'  DEC     R0          ; MORE TO STORE?
190 000116 003375 000000'  BGT     10$         ; BR IF YES
191 000120 000000 000000'  RETURN
192
193                                     ; ERRORS
194
195 000122 101$:  MSG$R  L0      ; ILLEGAL REMOTE DTE NAME
196
197                                     ; START OF REMOTE DTE ADDRESS
198
199
200 000130 012700 000017  RDTEST: MOV    #RDTLN,R0      ; GET LENGTH OF REMOTE DTE ADDRESS
201 000134 012701 000006'  MOV    #RDTE,R1      ; GET ADDRESS OF BUFFER
202 000140 105021 000000'  10$: CLRB   (R1)+    ; INITIALIZE BUFFER
203 000142 005300 000000'  DEC     R0          ; MORE TO INITIALIZE?
204 000144 003375 000000G  BGT     10$         ; BR IF YES
205 000146 105067 000000G  CLRB   COUNT      ; INITIALIZE DIGIT COUNT
206 000152 012767 000006' 000000G  MOV    #RDTE,NEXT    ; START STORING AT BEGINNING OF BUFFER
207 000160 000000 000000'  CALL    STRNXT      ; STORE CHARACTER
208 000164 000000 000000'  RETURN
209
210                                     ; PROCESS REMOTE DTE DIGIT
211
212
213 000166 026727 000000G 000024' RDTE: CMP    NEXT,#RDTEN    ; TOO MANY DIGITS?
214 000174 101003 000000'  BHI     101$      ; BR IF YES
215 000176 000000 000000'  CALL    STRNXT      ; STORE NEXT DIGIT
216 000202 000000 000000'  RETURN
217
218                                     ; ERRORS
219
220 000204 101$:  MSG$R  L1      ; ILLEGAL REMOTE DTE ADDRESS

```

```

119          .SBTTL TPARS STATE TABLES
120          ;
121          ; TPARS STATE TABLES
122          ;
123          ISTAT$ X29ST,X29KW
124          ;
125          ; X29$DF
126          ;
127          STATES$ X29DF
128          TRANS$ %X29$DF%,1,SYNERR
129          ;
130          STATES$
131          TRANS$ $NUMBR,,X29MXC ; MAXIMUM CIRCUITS
132          ;
133          STATES$
134          TRANS$ <','>
135          ;
136          STATES$
137          TRANS$ $NUMBR,,X29CTM ; COUNTER TIMER
138          ;
139          STATES$
140          TRANS$ !END,$EXIT,X29END
141          ;
142          ; CHECK FOR END OF SOURCE LINE
143          ;
144          STATES$ END
145          TRANS$ <','>,$EXIT
146          TRANS$ $EOS,$EXIT
147          ;
148          STATE$
  
```



```

143                                     .SBTTL  TPARS STATE TABLES
144                                     ;
145                                     ; TPARS STATE TABLES
146                                     ;
147 000060                             ISTAT$  X3PST,X3PKW
148
149                                     ;
150                                     ; X3P$DF
151                                     ;
152 000060                             STATES$ X3PDF
153 000060                             TRANS$  %X3P$DF%, , , 1,SYNERR
154
155 000060                             STATES$
156 000060                             TRANS$  $NUMBR, ,BLKDEF           ; DEFAULT BLOCK SIZE
157
158 000060                             STATES$
159 000060                             TRANS$  <','>
160
161 000060                             STATES$
162 000060                             TRANS$  $NUMBR, ,BLKMAX           ; MAXIMUM BLOCK SIZE
163
164 000060                             STATES$
165 000060                             TRANS$  <','>
166
167 000060                             STATES$
168 000060                             TRANS$  $NUMBR, ,WNDDEF           ; DEFAULT WINDOW SIZE
169
170 000060                             STATES$
171 000060                             TRANS$  <','>
172
173 000060                             STATES$
174 000060                             TRANS$  $NUMBR,$EXIT,WNDMAX       ; MAXIMUM WINDOW SIZE

```

```

130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147 000000
148 000000 005067 000000'
149 000004
150 000010 105777 000000G
151 000014 100420
152 000016 052010
153 000020 001422
154 000022
155 000026 103402
156 000030
157 000034
158 000040 005767 000000'
159 000044 001401
160 000046 000261
161 000050 016700 000000'
162 000054
163
164
165
166
167 000056 012767 000002 000000'
168 000064 000770
169 000066 012767 000006 000000'
170 000074 000757

;+
;CTBIO - CETAB I/O (RSX11M+ ONLY)
;
; THIS ROUTINE IS CALLED TO TRANSFER THE CETAB TASK IMAGE INTO CORE.
;
; INPUTS:
;   R1=FILE SPEC ADDRESS
;   R2=FILE SPEC LENGTH
;
; OUTPUTS:
;   C-BIT=SUCCESS/FAILURE
;   C=1
;       RO=ERROR CODE
;   C=0
;       R1 = CETAB ALLOCATION ADDRESS
;       R2 = CETAB ALLOCATION LENGTH
;
;CTBIO::
;   CLR      ERROR          ; NO ERROR, EITHER
;   CALL     $CLOPE         ; OPEN PROCESS FILE
;   TSTB     @ $CLIOS       ; LOOK AT GCL IOSB
;   BMI      101$           ; IF MI, FAILURE
;   BIS      (R0)+, (R0)    ; IS FILE CONTIGUOUS?
;   BEQ      111$           ; IF EQ, NO
;   CALL     PR.LBL         ; READ LABEL BLOCK
;   BCS      10$            ; IF CS, ERROR
;   CALL     PR.XFR         ; READ PROCESS TASK IMAGE INTO CORE
;   CALL     $CLDEQ         ; CLOSE PROCESS FILE
;   TST      ERROR          ; ANY ERRORS? (CLEARS C-BIT)
;   BEQ      30$            ; IF ZERO, NO
;   SEC      30$            ; INDICATE FAILURE
;   MOV      ERROR, R0      ; GET ERROR CODE
;   RETURN
;
; ERROR CONDITIONS
;
;101$: MOV      #ERR2, ERROR ; OPEN FAILURE
;      BR      20$
;111$: MOV      #ERR6, ERROR ; NOT CONTIGUOUS
;      BR      10$

```

ELTIME      CREATED BY    MACRO    ON 29-JUN-85 AT 17:42      PAGE 2      G 15  
MACRO CROSS REFERENCE      CREF    04.00  
MACRO NAME      REFERENCES  
RETURN      5-80



ERROR MESSAGES

|            |        |  |
|------------|--------|--|
| 97         | .SBTTL | ERROR MESSAGES   |
| 98         |        |  |
| 99         | .ENABL | LC   |
| 100        |        |  |
| 101 000016 | NTLERS | ,N2,10,CERR,RTSPC,CFLIN,<CUG name>                         |
| 102 000040 | NTLERS | ,N3,10,CERR,RTSPC,CFLIN,<CUG number>                       |
| 103 000064 | NTLERS | ,P4,10,CERR,RTSPC,CFLIN,<flags word>                       |
| 104 000110 | NTLERS | ,M1,8,CERR,RTSPC,CFLIN,<CUG name block allocation failure> |
| 105        |        |  |
| 106        | .DSABL | LC   |
| 107        | .LIST  | BEX  |
| 108        | .EVEN  |  |
| 109 000000 | .PSECT |  |

CFPDSA - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:15 <sup>H 2</sup>  
Table of contents

|     |     |                          |
|-----|-----|--------------------------|
| 5-  | 55  | MACRO DEFINITIONS        |
| 6-  | 62  | LOCAL SYMBOL DEFINITIONS |
| 7-  | 69  | ERROR MESSAGES           |
| 8-  | 87  | LOOK FOR DSA\$DF MACRO   |
| 9-  | 121 | TPARS STATE TABLES       |
| 10- | 163 | DSC\$DF STATE TABLES     |
| 11- | 188 | DSA\$DF ACTION ROUTINES  |
| 14- | 286 | DSC\$DF ACTION ROUTINES  |

|                    |                    |                  |                  |                     |  |
|--------------------|--------------------|------------------|------------------|---------------------|--|
| A\$\$CHK= 000000   | DSACUG= ***** GX   | G\$T\$TK= 000000 | N\$PEM= 000001   | \$EOS = 000012      |  |
| A\$\$CPS= 000000   | DSADF 000000R 003  | G\$SWRD= 000000  | PCKBCD= ***** GX | \$ERRMO 000106R 002 |  |
| A\$\$PRI= 000000   | DSAKW 000000RG 004 | I\$SRAR= 000000  | P\$P45= 000000   | \$ERRM2 000136R 002 |  |
| A\$TRP= 000000     | DSARC= ***** GX    | I\$SRDN= 000000  | P\$SWRD= 000000  | \$ERRM3 000166R 002 |  |
| CALVAL 000152R 003 | DSARDP= ***** GX   | K\$CNT= 177546   | Q\$OPT= 000010   | \$ERRN2 000000R 002 |  |
| CERR = ***** GX    | DSARDT= ***** GX   | K\$CSR= 177546   | RTSPC = ***** GX | \$ERRN8 000022R 002 |  |
| CFGBF = ***** GX   | DSASHI= ***** GX   | K\$SLDC= 000000  | R\$DER= 000000   | \$ERRN9 000056R 002 |  |
| CFGSZ = ***** GX   | DSASLO= ***** GX   | K\$TPS= 000074   | R\$SK11= 000001  | \$ERR1T= ***** GX   |  |
| CFLIN = ***** GX   | DSAST 000000RG 003 | LD\$LP = 000000  | R\$SND= 000000   | \$EXIT = 000000     |  |
| CHKSB 000254R      | DSCDF 000130R 003  | L\$ASG= 000000   | R\$11M= 000000   | \$FAIL = 177777     |  |
| COUNT = ***** GX   | DSCMCT= ***** GX   | L\$DRV= 000000   | SHI 000102R 003  | \$LAMD= 000000      |  |
| CUNMMX= ***** GX   | DSCMSK= ***** GX   | L\$P11= 000001   | SLO 000062R 003  | \$NUMBR= 000002     |  |
| C\$CKP= 000000     | DSCMX = 000040     | L\$11R= 000000   | STRHEX 000422R   | \$QDSA 000000RG     |  |
| C\$CORE= 000400    | DSCVAL= ***** GX   | M\$CRB= 000124   | STRNXT= ***** GX | \$QDSC 000006RG     |  |
| C\$SRSH= 177564    | DSCVCT= ***** GX   | M\$CRX= 000000   | SUBAMX= ***** GX | \$RAD50= 000016     |  |
| DACUG 000122R 003  | D\$BUG= 177514     | M\$FCS= 000000   | SYNERR= ***** GX | \$SAVRG= ***** GX   |  |
| DACUMN 000306R     | D\$ISK= 000000     | M\$MGE= 000000   | S\$WRG= 000000   | \$STRNG= 000004     |  |
| DARDEN= ***** GX   | D\$L11= 000001     | M\$NET= 000000   | S\$YSZ= 007600   | \$SUBXP= 000010     |  |
| DARDND 000142R     | D\$YNC= 000000     | M\$OVR= 000000   | T\$KMG= 000000   | \$SFLG= 177777      |  |
| DARDNO 000116R     | D\$YNM= 000000     | NEXT = ***** GX  | T\$MIN= 000000   | \$SKEY= 000001      |  |
| DARDNX 000032R 003 | END 000160R 003    | N\$ACC= 000001   | V\$CTR= 001000   | \$SSTA= 000000      |  |
| DARDST 000060R     | E\$XPR= 000000     | N\$BUF= 000001   | X\$DBT= 000000   | \$STMP= 000007R 005 |  |
| DARMLN= ***** GX   | FMT10 = ***** GX   | N\$LDV= 000001   | \$ALPHA= 000022  | .PNUMB= ***** GX    |  |
| DASSHI 000224R     | FM.10 = 000000     | N\$MCP= 000031   | \$ANY = 000020   | .PNUMH= ***** GX    |  |
| DASSLO 000176R     | F\$LVL= 000001     | N\$MLL= 000001   | \$BLANK= 000006  | .PSTCN= ***** GX    |  |
| DCMSK 000346R      | G\$TPP= 000000     | N\$MOV= 000010   | \$DIGIT= 000024  | .PSTPT= ***** GX    |  |
| DCVAL 000374R      | G\$TSS= 000000     | N\$NCT= 000001   | \$DNUMB= 000014  | .TPARS= ***** GX    |  |

. ABS. 000000 000 (RW,I,GBL,ABS,OVR)  
000540 001 (RW,I,LCL,REL,CON)  
DATA 000216 002 (RW,I,LCL,REL,CON)  
\$STATE 000170 003 (RW,D,LCL,REL,CON)  
\$KTAB 000004 004 (RW,D,LCL,REL,CON)  
\$KSTR 000016 005 (RW,D,LCL,REL,CON)  
Errors detected: 0

# \*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 12528 Words ( 49 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:37.26  
SY:CFPDSA.V2,[132,134]CFPDSA/CR/-SP=SY:[1,1]RSXCMC.SML/ML,[130,110]NETLIB/ML,[130,10]RSXCMC/PA:1,[132,10]CFPDSA

```

231
232
233
234
235
236 000530 012700 000000G
237 000534 000402
238
239
240
241 000536 012700 000000G
242 000542 005767 000000G
243 000546 001006
244 000550 105767 000001G
245 000554 001003
246 000556 116710 000000G
247 000562
248
249
250
251 000564 022700 000000G
252 000570 001403
253 000572
254 000600
255

; PRIORITY
;
; .ENABL LSB
DSPRI: MOV #DSTPRI,RO ; GET ADDRESS OF STORAGE FOR PRIORITY
        BR 10$ ; CONTINUE IN COMMON CODE

; OBJECT NUMBER
;
DSOBJ: MOV #DSOBJ,RO ; GET ADDRESS OF OBJECT NUMBER STORAGE
10$: TST .PNUMH ; LEGAL VALUE?
      BNE 101$ ; BR IF NO
      TSTB .PNUMB+1 ; IS IT A BYTE VALUE?
      BNE 101$ ; BR IF NO
      MOVB .PNUMB,(RO) ; STORE SPECIFIED VALUE
      RETURN

; ERRORS
;
101$: CMP #DSOBJ,RO ; IS THIS FOR OBJECT NUMBER?
      BEQ 102$ ; BR IF YES
      MSG$R N5 ; ELSE ILLEGAL PRIORITY
102$: MSG$R N6 ; ILLEGAL OBJECT NUMBER
      .DSABL LSB

```



CFPDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:17 <sup>H 5</sup>  
Table of contents

|     |     |                                      |
|-----|-----|--------------------------------------|
| 5-  | 55  | MACRO DEFINITIONS                    |
| 6-  | 79  | LOCAL SYMBOL DEFINITIONS             |
| 7-  | 99  | LOCAL DATA                           |
| 8-  | 121 | ERROR MESSAGES                       |
| 9-  | 135 | LOOK FOR DTE\$DF MACRO               |
| 10- | 164 | TPARS STATE TABLES                   |
| 11- | 216 | LINE-ID SUBEXPRESSION                |
| 12- | 239 | DTE\$DF ACTION ROUTINES              |
| 16- | 423 | CHKDTE - CHECK FOR VALID DTE ADDRESS |
| 17- | 464 | CHKDEV - CHECK VALIDITY OF LINE-ID   |
| 18- | 516 | BIASX - SET APR6 BIAS CORRECTLY      |

```

394 000600 011063 000044      MOV      (R0),L$TCLZ(R3)      ;; STORE TIME COUNTERS LAST ZEROED
395 000604 112763 000001 000003      MOV      #L$OFF,L$NMST(R3)      ;; STORE ASSUMED OFF STATE
396 000612 032767 000000G 000000G      BIT      #DE.ON,DTEFLG      ;; SET DTE STATE ON?
397 000620 001403      BEQ      60$      ;; BR IF NO- OFF IT IS
398 000622 112763 000000 000003      MOV      #L$ON,L$NMST(R3)      ;; STORE ON STATE
399 000630 116363 000003 000002 60$:      MOV      L$NMST(R3),L$OMST(R3)      ;; SET PREVIOUS STATE AS WELL
400 000636 010300      MOV      R3,R0      ;; COPY DTE ADDRESS
401 000640 062700 000122      ADD      #L$LEN,R0      ;; OFFSET TO START OF HASH TABLE
402 000644 010067 000000G      MOV      R0,HSHADD      ;; SAVE HASH TABLE ADDRESS
403 000650      CALL      BIASX      ;; SET CORRECT BIAS ON ADDRESS
404 000654 010063 000030      MOV      R0,L$CHTB(R3)      ;; STORE HASH TABLE ADDRESS
405 000660 016763 000000G 000032      MOV      HSHSZ,L$CTEN(R3)      ;; SAVE NUMBER OF ENTRIES
406 000666 017700 000000G      MOV      @PSIPT,R0      ;; GET ADDRESS OF PSI HOME BLOCK
407 000672 062700 000002      ADD      #H$LDTE,R0      ;; POINT TO DTE DESCRIPTOR LISTHEAD
408 000676 016705 000000G      MOV      DTEDES,R5      ;; GET UNMAPPED ADDR OF BLOCK TO LINK
409 000702      CALL      LNKEND      ;; LINK BLOCK AT END OF LIST
410 000706      RESMAP      ;; RESTORE PREVIOUS MAPPING
411 000712      RETURN      110$:
412      120$: BCS      103$      ; BR IF ALLOCATION ERROR
413 000714 103407      RETURN
414 000716
415      ;
416      ; ERRORS
417      ;
418 000720      101$: MSG$R 03      ; ILLEGAL DTE ADDRESS
419 000726      102$: MSG$R 07      ; LINE NOT IN SYSTEM
420 000734      103$: MSG$R 0P      ; ALLOCATION FAILURE
421 000742      104$: MSG$R 0Q      ; TOO MANY DTEs DEFINED

```

.TITLE CFPPSN - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 30-JAN-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/s V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL   | VALUE    | REFERENCES | 8-117   | 8-118   | 8-119   | 8-120   | 8-121   | 8-122  | 8-123  | 8-124 |
|----------|----------|------------|---------|---------|---------|---------|---------|--------|--------|-------|
| CERR     | = *****  | GX         | 8-117   | 8-118   | 8-119   | 8-120   | 8-121   | 8-122  | 8-123  | 8-124 |
| CFGFB    | = *****  | GX         | 9-147   |         |         |         |         |        |        |       |
| CFGGSZ   | = *****  | GX         | 9-146   |         |         |         |         |        |        |       |
| CFILIN   | = *****  | GX         | 8-117   | 8-118   | 8-119   | 8-120   | 8-121   | 8-122  | 8-123  | 8-124 |
| CHKSUB   | = 000276 | R          | 12-273  | #12-287 | 13-303  |         |         |        |        |       |
| FMT10    | = *****  | GX         | 8-117   | 8-118   | 8-119   | 8-120   | 8-123   | 8-124  |        |       |
| FMT8     | = *****  | GX         | 8-121   | 8-122   |         |         |         |        |        |       |
| FM.10    | = 000000 |            | #8-117  | #8-118  | #8-119  | #8-120  | #8-123  | #8-124 |        |       |
| FM.8     | = 000000 |            | #8-121  | #8-122  |         |         |         |        |        |       |
| HF\$DLM  | = 000002 |            | 12-277  | 13-308  | 13-353  |         |         |        |        |       |
| HS\$FLG  | = 000000 |            | *13-352 |         |         |         |         |        |        |       |
| HS\$HITS | = 000034 |            | *13-356 |         |         |         |         |        |        |       |
| HS\$LEN  | = 000042 |            | 13-312  | 13-317  |         |         |         |        |        |       |
| HS\$LOTS | = 000032 |            | *13-355 |         |         |         |         |        |        |       |
| HS\$NETW | = 000024 |            | 13-321  |         |         |         |         |        |        |       |
| HS\$NPT  | = 000022 |            | *13-351 |         |         |         |         |        |        |       |
| HS\$PTB  | = 000020 |            | *13-350 |         |         |         |         |        |        |       |
| KSAR5    | = *****  | GX         | 13-338  | 13-346  |         |         |         |        |        |       |
| NS\$VCT  | = *****  |            | 13-335  |         |         |         |         |        |        |       |
| PNAMMX   | = 000006 |            | #6-82   | 11-218  | 13-323  |         |         |        |        |       |
| PORT     | = 000160 | R          | #11-244 |         |         |         |         |        |        |       |
| PORTMX   | = 000400 |            | #6-83   | 11-248  |         |         |         |        |        |       |
| PORTNO   | = *****  | GX         | *11-250 | 13-327  | 13-342  | 13-351  |         |        |        |       |
| PSIPT    | = *****  | GX         | 11-215  | 13-330  |         |         |         |        |        |       |
| PSNDF    | = 000000 | R          | 9-143   |         |         |         |         |        |        |       |
| PSNEND   | = 000356 | R          | #13-312 |         |         |         |         |        |        |       |
| PSNFG    | = 000216 | R          | #11-260 |         |         |         |         |        |        |       |
| PSNFLG   | = 000006 | R          | #7-109  | *11-231 | *11-262 | *12-277 | *13-308 | 13-352 | 13-353 |       |
| PSNHI    | = 000012 | R          | #7-111  | *11-233 | *12-276 | *13-307 | 13-356  |        |        |       |
| PSNKW    | = 000000 | RG         | 9-145   | #10-164 |         |         |         |        |        |       |
| PSNLO    | = 000010 | R          | #7-110  | *11-232 | *12-275 | 13-305  | 13-355  |        |        |       |
| PSNNAM   | = 000000 | R          | #7-108  | 11-221  | 13-322  |         |         |        |        |       |
| PSNNM    | = 000052 | R          | #11-215 |         |         |         |         |        |        |       |
| PSNSHI   | = 000330 | R          | #13-303 |         |         |         |         |        |        |       |
| PSNSLO   | = 000242 | R          | #12-273 |         |         |         |         |        |        |       |
| PSNST    | = 000000 | RG         | #10-164 |         |         |         |         |        |        |       |
| RTSPC    | = *****  | GX         | 8-117   | 8-118   | 8-119   | 8-120   | 8-121   | 8-122  | 8-123  | 8-124 |
| R\$R0    | = 000002 |            | #6-89   |         |         |         |         |        |        |       |
| R\$R1    | = 000004 |            | #6-90   |         |         |         |         |        |        |       |
| R\$R2    | = 000006 |            | #6-91   | *13-337 |         |         |         |        |        |       |
| R\$R3    | = 000010 |            | #6-92   |         |         |         |         |        |        |       |
| R\$R4    | = 000012 |            | #6-93   |         |         |         |         |        |        |       |
| R\$R5    | = 000014 |            | #6-94   |         |         |         |         |        |        |       |
| SPACE    | = 000040 |            | #6-84   | 11-223  |         |         |         |        |        |       |
| SUBAMX   | = *****  | GX         | 12-291  |         |         |         |         |        |        |       |
| SYNERR   | = *****  | GX         | *9-148  | *9-151  |         |         |         |        |        |       |
| \$BAS    | = *****  |            | 8-117   | 8-117   | 8-118   | 8-119   | 8-121   | 8-122  | 8-123  | 8-124 |
|          |          |            | 8-121   | 8-122   | 8-122   | 8-123   | 8-124   |        |        |       |
| \$ALL16  | = *****  | GX         | 13-313  |         |         |         |         |        |        |       |
| \$ALPHA  | = 000022 |            | #10-164 |         |         |         |         |        |        |       |
| \$ANY    | = 000020 |            | #10-164 |         |         |         |         |        |        |       |
| \$BLANK  | = 000006 |            | #10-164 |         |         |         |         |        |        |       |

```

286                                     .SBTTL PVC$DF ACTION ROUTINES
287
288                                     : PVC CIRCUIT ID
289
290
291 000056 016702 000000G   PVNAM: MOV   .PSTCN,R2   ; GET NUMBER OF CHARACTERS FOUND
292 000062 022702 000006   CMP   #PVNAMX,R2   ; LEGAL PVC CIRCUIT ID?
293 000066 103432          BLO   101$          ; BR IF NO
294 000070 012700 000024'   MOV   #PVCNAM,R0   ; POINT TO STORAGE FOR CIRCUIT ID
295 000074 012701 000006   MOV   #PVNAMX,R1   ; GET LENGTH OF CIRCUIT ID
296 000100 112720 000040   10$: MOVB  #SPACE,(R0)+ ; INITIALIZE NAME TO SPACES
297 000104 005301          DEC   R1             ; MORE TO INITIALIZE?
298 000106 003374          BGT   10$           ; BR IF YES
299 000110 016700 000000G   MOV   .PSTPT,R0   ; POINT TO START OF CIRCUIT ID
300 000114 012701 000024'   MOV   #PVCNAM,R1   ; POINT TO STORAGE AREA FOR CIRCUIT ID
301 000120 112021          20$: MOVB  (R0)+,(R1)+ ; STORE CIRCUIT ID
302 000122 005302          DEC   R2             ; MORE TO STORE?
303 000124 003375          BGT   20$           ; BR IF YES
304 000126 016767 000000G 000006'   MOV   BLKSZD,PVCPSZ ; SET UP DEFAULT MAXIMUM BLOCK SIZE
305 000134 016767 000000G 000010'   MOV   WNDSDZ,PVCWSZ ; SET UP DEFAULT WINDOW SIZE
306 000142 005067 000012'   CLR   PVCOWN    ; INITIALIZE PROCESS OWNER
307 000146 105067 000023'   CLRB  PVCFLG    ; INITIALIZE FLAGS BYTE
308 000152
309
310                                     : ERRORS
311
312 000154   101$: MSG$R NO          ; ILLEGAL CIRCUIT ID
313
314                                     : LOGICAL CHANNEL NUMBER
315
316
317 000162 005767 000000G   PVLCN: TST   .PNUMH    ; LEGAL VALUE?
318 000166 001011          BNE   101$          ; BR IF NO
319 000170 016700 000000G   MOV   .PNUMB,R0   ; GET SPECIFIED LOGICAL CHANNEL NUMER
320 000174 003406          BLE   101$          ; BR IF ILLEGAL VALUE
321 000176 020027 000000G   CM2  R0,#CHNLMX ; LEGAL VALUE?
322 000202 101003          BHI   101$          ; BR IF NO
323 000204 010067 000002'   MOV   R0,PVLCN   ; STORE LOGICAL CHANNEL NUMBER
324 000210
325
326                                     : ERRORS
327
328 000212   101$: MSG$R 09          ; ILLEGAL LOGICAL CHANNEL NUMBER
329
330                                     : COUNTER TIMER
331
332
333 000220 005767 000000G   PVCT: TST   .PNUMH    ; LEGAL VALUE?
334 000224 001004          BNE   101$          ; BR IF NO
335 000226 016767 000000G 000004'   MOV   .PNUMB,PVCCTM ; SAVE COUNTER TIMER VALUE
336 000234
337
338                                     : ERRORS
339
340 000236   101$: MSG$R 05          ; ILLEGAL COUNTER TIMER VALUE

```

CFPPVC CREATED BY MACRO ON 29-JUN-85 AT 00:19 PAGE 2 H 10  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL    | VALUE      | REFERENCES                                |
|-----------|------------|---|
| PVCNAM    | 000024 R   | #7-140 11-294 11-300 13-418 15-507 20-674 |
| PVCOWN    | 000012 R   | #7-133 *11-306 *12-375 14-472             |
| PVCPRT    | 000022 R   | #7-138 *16-547 17-585 18-637              |
| PVCPSZ    | 000006 R   | #7-131 *11-304 *12-348 17-582             |
| PVCST     | 000000 RG  | #10-200                                   |
| PVCT      | 000220 R   | #11-333                                   |
| PVCTL     | 001544 R   | #20-700                                   |
| PVCWSZ    | 000010 R   | #7-132 *11-305 *12-360 17-581             |
| PVDLM     | 001446 R   | #20-674                                   |
| PVEND     | 000404 R   | 12-390 #13-401                            |
| PVFG      | 000346 R   | #12-385                                   |
| PVLGN     | 000162 R   | #11-317                                   |
| PVNAM     | 000056 R   | #11-291                                   |
| PVNAMX    | = 000006   | #6-103 11-292 11-295 13-417 15-506        |
| PVOWN     | 000310 R   | #12-370                                   |
| PVPSZ     | 000244 R   | #12-346                                   |
| PVSTA     | 001554 R   | #20-702                                   |
| PVWSZ     | 000266 R   | #12-358                                   |
| RTSPC     | = ***** GX | 8-156 8-157 8-158 8-159                   |
| R\$R0     | = 000002   | #6-108                                    |
| R\$R1     | = 000004   | #6-109                                    |
| R\$R2     | = 000006   | #6-110                                    |
| R\$R3     | = 000010   | #6-111                                    |
| R\$R4     | = 000012   | #6-112                                    |
| R\$R5     | = 000014   | #6-113                                    |
| SPACE     | = 000040   | #6-102 11-296                             |
| SP\$P4    | 000036 RG  | #7-149 17-595                             |
| SYNERR    | = ***** GX | *9-184 *9-187                             |
| S\$B\$BAS | = *****    | 8-156 8-157 8-158 8-159 8-159 8-159       |
| W\$D\$SZD | = ***** GX | 11-305                                    |
| XT\$PVC   | = 000001   | 17-596                                    |
| X\$A\$PQ  | 000076     | *17-593 17-594 *17-594                    |
| X\$C\$TIM | 000060     | *17-583                                   |
| X\$D\$TE  | 000024     | *17-584                                   |
| X\$LCN    | 000026     | *17-586                                   |
| X\$LEN    | 000106     | 14-461 17-574                             |
| X\$MOWN   | 000023     | *14-475                                   |
| X\$PKSZ   | 000016     | *17-582                                   |
| X\$PRT    | 000021     | *17-585                                   |
| X\$RXQ    | 000072     | *17-589 17-590 *17-590                    |
| X\$ST     | 000005     | *17-595                                   |
| X\$TCLZ   | 000030     | *14-478                                   |
| X\$TXQ    | 000066     | *17-587 17-588 *17-588                    |
| X\$TYP    | 000015     | *17-596                                   |
| X\$WAQ    | 000062     | *17-591 17-592 *17-592                    |
| X\$WSZ    | 000020     | *17-581                                   |
| ZTIME     | = ***** GX | 14-476                                    |
| \$ALPHA   | = 000022   | #10-200                                   |
| \$ANY     | = 000020   | #10-200                                   |
| \$BLANK   | = 000006   | #10-200                                   |
| \$CAT5    | = ***** GX | 12-374 20-675                             |
| \$CEACX   | = ***** GX | 13-413 14-466 15-504 16-538 17-572 18-617 |

```

222
223
224
225
226 000212 004567 000000G
227 000216 012700 000010
228 000222 012704 000006
229 000226 012705 000025
230 000232
231 000236
232 000242 103472
233 000244 012701 000024
234
235 000250
236 000254
237 000260 103006
238 000262
239 000274 000452
240 000276
241 000302 010005
242 000304 010046
243 000306
244 000312 012600
245 000314 010002
246 000316 105022
247 000320 005301
248 000322 003375
249 000324 010001
250 000326 062701 000002
251 000332 012702 000000
252 000336 012703 000006
253 000342 112221
254 000344 005303
255 000346 003375
256 000350 116760 000000G 000013
257 000356 012701 000010
258 000362 012702 000025
259 000366 010003
260 000370 062703 000014
261 000374 112223
262 000376 005301
263 000400 003375
264 000402 017700 000000G
265 000406 062700 000004
266 000412
267 000416
268 000422
269
270 000424 103404
271 000426
272
273
274
275 000430
276 000436

; END OF REMOTE DTE ADDRESS
RDEND: JSR R5,$SAVRG ; SAVE R3-R5
MOV #<RDTLN+1>/2,R0 ; GET LENGTH OF PACKED ADDRESS
MOV #RDE,R4 ; GET ADDRESS OF REMOTE DTE ADDR BUFFER
MOV #RDEP,R5 ; GET ADDRESS OF PACKED ADDRESS BUFFER
CALL PCKBCD ; PACK ADDRESS IN BCD FORMAT
CALL CHKRDT ; CHECK DTE NAME FOR UNIQUENESS
BCS 101$ ; BR IF NOT UNIQUE
MOV #R$LEN,R1 ; GET LENGTH OF BLOCK TO ALLOCATE

; ENTER SYSTEM STATE
CALL $XALOC ; ALLOCATE FROM NETWORK POOL
BCC 10$ ; BR IF SUCCESSFUL ALLOCATION
RET C R0 ; ELSE SET USER C-BIT
BR 40$ ; AND EXIT

10$: SAVMAP ; SAVE CURRENT MAPPING
MOV R0,R5 ; SAVE UNMAPPED ADDRESS
MOV R0, (SP) ; SET UNMAPPED ADDRESS FOR CONVERSION
CALL $CEACK ; CONVERT TO MAPPED ADDRESS
MOV (SP)+,R0 ; RETRIEVE MAPPED ADDRESS
MOV R0,R2 ; GET ADDRESS OF BLOCK
15$: CLRB (R2)+ ; INITIALIZE REMOTE DTE NAME BLOCK
DEC R1 ; MORE TO INITIALIZE?
BGT 15$ ; BR IF YES
MOV R0,R1 ; GET ADDRESS OF BLOCK
ADD #R$NAM,R1 ; POINT TO REMOTE DTE NAME
MOV #RDTNAM,R2 ; POINT TO SPECIFIED NAME
MOV #RDTMX,R3 ; GET LENGTH OF NAME
20$: MOVB (R2)+,(R1)+ ; STORE REMOTE DTE NAME
DEC R3 ; MORE TO STORE?
BGT 20$ ; BR IF YES
MOVB COUNT,R$DAL(R0) ; STORE LENGTH OF DTE ADDRESS
MOV #<RDTLN+1>/2,R1 ; GET LENGTH OF DTE ADDRESS
MOV #RDEP,R2 ; GET ADDRESS OF REMOTE DTE ADDRESS
MOV R0,R3 ; POINT TO REMOTE DTE NAME BLOCK
ADD #R$DTE,R3 ; POINT TO DTE ADDRESS FIELD
30$: MOVB (R2)+,(R3)+ ; STORE REMOTE DTE ADDRESS
DEC R1 ; MORE TO STORE?
BGT 30$ ; BR IF YES
MOV @PSIPT,R0 ; GET ADDRESS OF PSI HOME BLOCK
ADD #H$RDE,R0 ; POINT TO REMOTE DTE LISTHEAD
CALL LNKEND ; LINK AT END OF LIST
40$: RESMAP ; EXIT SYSTEM STATE
RETURN

50$: BCS 102$ ; BR IF ALLOCATION FAILED
RETURN

; ERRORS
101$: MSG$R L0 ; ILLEGAL REMOTE DTE NAME
102$: MSG$R L2 ; REMOTE DTE BLOCK ALLOCATION FAILURE

```

```

150 .SBTTL X29$DF ACTION ROUTINES
151
152 ;
153 ; MAXIMUM CIRCUITS
154
155 000052 005767 000000G X29MXC: TST .PNUMH ; SINGLE WORD VALUE?
156 000056 001007 000000G BNE 101$ ; NO- ILLEGAL
157 000060 105767 000001G TSTB .PNUMB+1 ; SINGLE BYTE VALUE?
158 000064 001004 000000G BNE 101$ ; NO- ILLEGAL
159 000066 116767 000000G 000206' MOVB .PNUMB,MAXVC ; SAVE MAXIMUM CIRCUITS VALUE
160 000074 RETURN ; RETURN
161
162 000076 101$: MSG$R 9B ; ILLEGAL MAX CIRCUITS VALUE
163
164 ;
165 ; COUNTER TIMER
166
167 000104 005767 000000G X29CTM: TST .PNUMH ; SINGLE WORD VALUE?
168 000110 001004 000000G BNE 101$ ; NO- ILLEGAL
169 000112 016767 000000G 000210' MOV .PNUMB,CTIMR ; SAVE COUNTER TIMER VALUE
170 000120 RETURN ; RETURN
171
172 000122 101$: MSG$R 9C ; ILLEGAL COUNTER TIMER VALUE
173
174 ;
175 ; END OF X29$DF MACRO
176
177 000130 017700 000000G X29END: MOV @PSIPT,R0 ; POINT TO HOME BLOCK
178 000134 001437 000000G BEQ 101$ ; THIS, OF COURSE, CAN NEVER HAPPEN
179 000136 005760 000040 TST H$X29C(R0) ; X.29 DATABASE ALREADY EXIST?
180 000142 001034 000000G BNE 101$ ; YES- ILLEGAL 2ND OCCURRENCE
181
182 000144 012701 000016 MOV #Q$LEN,R1 ; GET LENGTH OF X.29 DATABASE
183 000150 $ALL16 CALL $ALL16 ; ALLOCATE FROM DSR
184 000154 103432 BCS 102$ ; BR IF UNABLE TO ALLOCATE
185 000156 010002 MOV R0,R2 ; SAVE POINTER TO BLOCK
186 000160 012701 000007 MOV #Q$LEN/2,R1 ; GET LENGTH IN WORDS
187 000164 005020 10$: CLR (R0)+ ; ZERO THE BLOCK
188 000166 005301 DEC R1 ; MORE TO ZERO?
189 000170 001375 BNE 10$ ; YES, KEEP GOING
190
191 000172 016762 000206' 000002 MOV MAXVC,Q$MXVC(R2) ; STORE MAXIMUM CIRCUITS VALUE
192 000200 016762 000210' 000000 MOV CTIMR,Q$CTIM(R2) ; STORE COUNTER TIMER VALUE
193
194 000206 016700 000000G MOV ZTIME,R0 ; GET $ZTIME ADDRESS
195 000212 062700 000002 ADD #2,R0 ; $ZTIME+2
196 000216 011062 000010 MOV (R0),Q$TCLZ(R2) ; STORE TIME COUNTERS LAST ZEROED
197 000222 017700 000000G MOV @PSIPT,R0 ; GET HOME BLOCK POINTER
198 000226 010260 000040 MOV R2,H$X29C(R0) ; STORE POINTER IN HOME BLOCK
199 000232 RETURN ; RETURN
200
201 000234 101$: MSG$R 9A ; ILLEGAL OCCURRENCE OF X29$DF
202 000242 102$: MSG$R 9D ; X.29 BLOCK ALLOCATION FAILURE
203
204 000001 .END

```



```

176 .SBTTL CHN$DF STATE TABLE
177
178 ; CHN$DF STATE TABLE
179 ;
180 ;
181 STATES CHNDF
182 TRANS %CHN$DF%,1,SYNERR
183
184 STATES
185 TRANS $NUMBR,,CHLO ; LOW END OF CHANNEL RANGE
186
187 STATES
188 TRANS :END,$EXIT,CHEND
189 TRANS <'>
190
191 STATES
192 TRANS $NUMBR,END,CHHI ; HIGH END OF CHANNEL RANGE
193
194 ;
195 ; CHECK FOR END OF SOURCE LINE
196 ;
197 STATES END
198 TRANS <'>,$EXIT
199 TRANS $EOS,$EXIT
200
201 STATES

```

```

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190 000076 012700 000000'
191 000102
192 000106 103441
193 000110 032760 040000 000030
194 000116 001444
195 000120 016001 000016
196 000124 026001 000020
197 000130 001037
198 000132 022701 000200
199 000136 103444
200 000140 016002 000352
201 000144 060102
202 000146 022702 000200
203 000152 103442
204 000154
205 000170 010167 000000G
206 000174 005002
207
208 000176
209 000202 103416
210 000204 010102
211 000206 010001
212 000210
213
214
215
216 000212 116777 000000G 000000G 101$: MOV $IOSB,$CLIOS ; COPY ERROR CODE
217 000220 012767 000010 000000' MOV #ERR8,ERROR ; LABEL BLOCK READ FAILURE
218 000226 000417 BR 142$
219 000230 012767 000012 000000' 111$: MOV #ERR10,ERROR ; INVALID CEX IMAGE
220 000236 000413 BR 142$
221 000240 012767 000044 000000' 121$: MOV #ERR36,ERROR ; CETAB ALLOCATION FAILURE
222 000246 000407 BR 142$
223 000250 012767 000016 000000' 131$: MOV #ERR14,ERROR ; CEX EXCEEDS 4K WORDS
224 000256 000403 BR 142$
225 000260 012767 000020 000000' 141$: MOV #ERR16,ERROR ; CEX EXTENSION EXCEEDS 4K WORDS
226 000266 000261 142$: SEC ; INDICATE FAILURE
227 000270 RETURN

```

PR.LBL - READ LABEL BLOCK

THIS ROUTINE IS CALLED TO READ THE CETAB LABEL BLOCK AND TO VERIFY THAT THE FILE IS A VALID CETAB IMAGE. ON EXIT, THE LOGICAL BLOCK I/O POINTERS ARE SETUP TO ACCESS THE FIRST TEXT BLOCK.

INPUTS:  
NONE

OUTPUTS:  
C-BIT=SUCCESS/FAILURE  
R1=CETAB ALLOCATION ADDRESS  
R2=CETAB ALLOCATION LENGTH  
R0=DESTROYED  
\$LBN=FIRST TEXT BLOCK LOGICAL BLOCK NUMBER

PR.LBL: MOV #DSKBUF,R0 ; GET BUFFER ADDRESS  
CALL \$RLBL ; READ LABEL BLOCK  
BCS 101\$ ; IF CS, FAILURE  
BIT #T\$NHD,\$BFLG(R0) ; DOES CETAB HAVE A HEADER?  
BEQ 111\$ ; IF EQ, YES  
MOV \$BLDZ(R0),R1 ; GET LOAD SIZE IN BLOCKS  
CMP \$BMXZ(R0),R1 ; IS CETAB OVERLAID?  
BNE 111\$ ; IF NE, YES  
CMP #200,R1 ; IS CETAB > 4K WORDS?  
BLO 131\$ ; IF LO, YES  
MOV \$BEXT(R0),R2 ; GET EXTEND SIZE  
ADD R1,R2 ; ADD CETAB SIZE TO EXTEND SIZE  
CMP #200,R2 ; IS CETAB+EXTENSION > 4K WORDS?  
BLO 141\$ ; IF LO, YES  
ASL\$ 6,R1 ; CONVERT FROM BLOCKS TO BYTES  
MOV R1,\$LLEN ; SET THE TRANSFER SIZE  
CLR R2 ; SUCCESS INDICATOR

CALL \$ALL16 ; TRY TO ALLOCATE ROOM FOR CETAB  
BCS 121\$ ; IF CS, COULD NOT  
MOV R1,R2 ; COPY THE CETAB LENGTH  
MOV R0,R1 ; COPY THE CETAB ADDRESS  
RETURN ; RETURN TO CALLER

ERROR CONDITIONS

\*\*FILE\*\*ID\*\*LBNIO

```

LL      BBBB BBBB  NN  NN  IIIIII  000000
LL      BBBB BBBB  NN  NN  IIIIII  000000
LL      BB      BB  NN  NN  II      00      00
LL      BB      BB  NN  NN  II      00      00
LL      BB      BB  NNNN  NN  II      00      00
LL      BB      BB  NNNN  NN  II      00      00
LL      BBBB BBBB  NN  NN  NN  II      00      00
LL      BBBB BBBB  NN  NN  NN  II      00      00
LL      BB      BB  NN  NN  NNNN  II      00      00
LL      BB      BB  NN  NN  NNNN  II      00      00
LL      BB      BB  NN  NN  NN  II      00      00
LL      BB      BB  NN  NN  NN  II      00      00
LLLLLLLLLL BBBB BBBB  NN  NN  IIIIII  000000
LLLLLLLLLL BBBB BBBB  NN  NN  IIIIII  000000

```

```

....
....
....
....

```

```

LL      SSSSSSSS  TTTT TTTT
LL      SSSSSSSS  TTTT TTTT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LL      SSSSSS  TT
LL      SSSSSS  TT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LL      SS      TT
LLLLLLLLLL SSSSSSSS  TT
LLLLLLLLLL SSSSSSSS  TT

```

LDUIC - FIND THE LOAD UIC FOR N MACRO V05.03b Saturday 29-Jun-85 00:21 <sup>H 16</sup>  
Table of contents

5- 55 LOCAL DATA  
6- 77 CREATE "LOAD UIC STRING"

```

111                                     .SBTTL LOOK FOR CUG$DF MACRO
112
113                                     ;+
114                                     ; $QCUG - LOOK FOR CUG$DF MACRO
115                                     ;
116                                     ; INPUTS:
117                                     ;     NONE
118                                     ;
119                                     ; OUTPUTS:
120                                     ;     C-BIT=SUCCESS/FAILURE
121                                     ;     R3,R4,R5=DESTROYED
122                                     ; -
123 000000 012705 000000' $QCUG:: MOV    #CUGDF,R5    ; STATE TABLE ADDRESS
124 000004 005001          CLR    R1                ; FULL KEYWORD MATCH LENGTH
125 000006 012702 000000' MOV    #CUGKW,R2          ; KEYWORD TABLE ADDRESS
126 000012 016703 000000G MOV    CFGSZ,R3          ; RECORD LENGTH
127 000016 012704 000000G MOV    #CFGBF,R4          ; RECORD BUFFER ADDRESS
128 000022 005067 000000G CLR    SYNERR          ; CLEAR SYNTAX ERROR FLAG
129 000026          CALL    TPARS          ; GO DO THE PARSE
130 000032 103003          BCC    20$           ; IF CC, FOUND WHAT WE WERE LOOKING FOR
131 000034 005367 000000G DEC    SYNERR          ; DID SYNTAX ERROR OCCUR?
132 000040 001401          BEQ    101$          ; IF EQ, YES
133 000042          20$:  RETURN
134
135                                     ;
136                                     ; ERROR CONDITION
137                                     ;
138 000044          101$:  MSG$R 1T            ; SYNTAX ERROR
  
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

.TITLE CFPDSA - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDFNT HISTORY:

- 1.00 31-MAR-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

CFPDSA      CREATED BY    MACRO    ON 29-JUN-85 AT 00:16      PAGE 1      1 3

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES   |
|---------|-------------|--|
| CERR    | = ***** GX  | 7-75      7-76      7-77      7-78      7-79      7-80       |
| CFGBF   | = ***** GX  | 8-107  |
| CFGSSZ  | = ***** GX  | 8-106  |
| CFLIN   | = ***** GX  | 7-75      7-76      7-77      7-78      7-79      7-80       |
| CHKSB   | = 000254 R  | 12-229      #12-257  |
| COUNT   | = ***** GX  | *11-198  |
| CUNMMX  | = ***** GX  | 13-273   |
| DACUNM  | = 000306 R  | #13-272  |
| DARDEN  | = ***** GX  | 11-206   |
| DARDND  | = 000142 R  | #11-218  |
| DARDNO  | = 000116 R  | #11-206  |
| DARDST  | = 000060 R  | #11-193  |
| DARMLN  | = ***** GX  | 11-193      11-219   |
| DASSHI  | = 000224 R  | #12-243  |
| DASSLO  | = 000176 R  | #12-229  |
| DCMSK   | = 000346 R  | #14-291  |
| DCVAL   | = 000374 R  | #14-304  |
| DSACUG  | = ***** GX  | 13-275   |
| DSADF   | = 000000 R  | 8-101  |
| DSAKW   | = 000000 RG | 8-105      #9-125  |
| DSARCT  | = ***** GX  | *11-223  |
| DSARDP  | = ***** GX  | 11-221   |
| DSARDT  | = ***** GX  | 11-194      11-199      11-220                               |
| DSASHI  | = ***** GX  | *12-232      *12-247   |
| DSASLO  | = ***** GX  | *12-231      12-245  |
| DSAST   | = 000000 RG | #9-125   |
| DSCDF   | = 000130 R  | 8-103  |
| DSCMCT  | = ***** GX  | 14-292   |
| DSCMSK  | = ***** GX  | 14-291   |
| DSCMX   | = 000040    | #6-67      14-316  |
| DSCVAL  | = ***** GX  | 14-304   |
| DSCVCT  | = ***** GX  | 14-305   |
| FMT10   | = ***** GX  | 7-75      7-76      7-77      7-78      7-79      7-80       |
| FM.10   | = 000000    | #7-75      #7-76      #7-77      #7-78      #7-79      #7-80 |
| ISSAS   | = *****     | 14-334   |
| NEXT    | = ***** GX  | *11-199      11-206  |
| PCKBCD  | = ***** GX  | 11-222   |
| RTSPC   | = ***** GX  | 7-75      7-76      7-77      7-78      7-79      7-80       |
| RSEIS   | = *****     | 14-334   |
| RSE11D  | = *****     | 14-334   |
| STRHEX  | = 000422 R  | 14-293      14-306      #14-314                              |
| STRNXT  | = ***** GX  | 11-200      11-208   |
| SUBAMX  | = ***** GX  | 12-261   |
| SYNERR  | = ***** GX  | *8-108      *8-111   |
| SSBAS   | = *****     | 7-75      7-76      7-77      7-78      7-79                 |
|         |             | 7-79      7-80      7-80                                     |
| \$ALPHA | = 000022    | #9-125   |
| \$ANY   | = 000020    | #9-125   |
| \$BLANK | = 000006    | #9-125   |
| \$DIGIT | = 000024    | #9-125   |
| \$DNUMB | = 000014    | #9-125   |
| \$EOS   | = 000012    | #9-125   |

```

257
258
259      ; TASK NAME
260
261 000606 022767 000006 000000G DSTSK: CMP    #DSTKMX,.PSTCN      ; LEGAL NUMBER OF CHARACTERS?
262 000614 103420          BLO    101$                          ; BR IF NO
263 000616          SAVRG  <R3>                                ; SAVE R3
264 000620 012703 000000G      MOV    #DSTTSK,R3                ; POINT TO STORAGE FOR TASK NAME
265 000624 016700 000000G      MOV    .PSIPT,R0                 ; GET START OF STRING
266 000630 012701 000001      10$:  MOV    #1,R1                 ; PERIOD IS A VALID CHARACTER
267 000634          CALL   $CATS                                ; CONVERT TO RAD50
268 000640 010123          MOV    R1,(R3)+                       ; STORE TASK NAME
269 000642 162767 000003 000000G      SUB    #3,.PSTCN          ; MORE TO CONVERT?
270 000650 003367          BGT    10$                            ; BR IF YES
271 000652          RESRG  <R3>                                ; RESTORE R3
272 000654          RETURN
273
274      ; ERRORS
275
276 000656      101$:  EMSG$R  N7                                ; ILLEGAL TASK NAME

```



.TITLE CFPDTE - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1981, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 30-JAN-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462

.SBTTL CHKDTE - CHECK FOR VALID DTE ADDRESS

```

+
CHKDTE - CHECK FOR VALID DTE ADDRESS
INPUTS:
  NONE
OUTPUTS:
  CARRY CLEAR - DTE ADDRESS IS UNIQUE
  R4 = COUNT OF DTEs DEFINED ALREADY
  CARRY SET - DTE ADDRESS IS NOT UNIQUE
  R0,R1,R2 DESTROYED
-
CHKDTE: MOV    @PSIPT,R0      ; POINT TO PSI HOME BLOCK
        ADD    #H$LDTE,R0    ; POINT TO LOCAL DTE LISTHEAD

        SWSTK$ 40$           ;; ENTER SYSTEM STATE
        SAVMAP           ;; SAVE CURRENT MAPPING
        CLR    R4           ;; INITIALISE DTE COUNT
10$:    MOV    (R0),R0        ;; POINT TO NEXT BLOCK IN LIST
        BEQ    30$          ;; BR IF END OF LIST
        INC    R4           ;; ONE MORE BLOCK SEEN
        MOV    R0,-(SP)      ;; SET UP UNMAPPED ADDRESS
        CALL   $CEACK        ;; CONVERT TO MAPPED ADDRESS
        MOV    (SP)+,R0      ;; RETRIEVE MAPPED ADDRESS
        MOV    #<DTEALN+1>/2,R1 ;; GET DTE ADDRESS LENGTH
        MOV    R0,R2        ;; GET ADDRESS OF LOCAL DTE DESCRIPTOR
        ADD    #L$DTEA,R2    ;; POINT TO DTE ADDRESS
        MOV    #DTEPC,R3     ;; POINT TO SPECIFIED DTE ADDRESS
20$:    CMPB   (R2)+,(R3)+    ;; DO THE ADDRESSES MATCH?
        BNE    10$          ;; BR IF NO: UNIQUE SO FAR
        DEC    R1           ;; MORE OF ADDRESS TO CHECK?
        BGT    20$          ;; BR IF YES
        RETC   R0           ;; INDICATE DUPLICATE ADDRESS
30$:    RESMAP           ;; RESTORE PREVIOUS MAPPING
        MOV    R4,R$R4(SP)   ;; RETURN COUNT IN TASK R4
40$:    RETURN
  
```

000750 017700 000000G  
 000754 062700 000002  
 000760  
 000764  
 000770 005004  
 000772 011000  
 000774 001425  
 000776 005204  
 001000 010046  
 001002  
 001006 012600  
 001010 012701 000010  
 001014 010002  
 001016 062702 000020  
 001022 012703 000017  
 001026 122223  
 001030 001360  
 001032 005301  
 001034 003374  
 001036  
 001050  
 001054 010466 000012  
 001060

```

55      .SBTTL  MACRO DEFINITIONS
56
57      ;
58      ; LIBRARY MACROS
59      ;
60      .MCALL  PHBDF$,EMSG$,RETC,SAVRG,RESRG,NTLR$
61      .MCALL  ISTAT$,STAT$,TRAN$
62
63      ;
64      ; LIBRARY SYMBOLS
65      ;
66      000000      PHBDF$          ; DEFINE PSI HOME BLOCK OFFSETS
67
68      ;
69      ; LOCAL MACRO DEFINITIONS
70      ;
71      .MACRO  SAVMAP
72      MOV     @KSAR5,-(SP)      ; SAVE APR 5
73      .ENDM
74
75      .MACRO  RESMAP
76      MOV     (SP)+,@KSAR5      ; RESTORE APR5
77      .ENDM
78

```

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL    | VALUE      | REFERENCES           |
|-----------|------------|----------------------|
| \$CEACX   | = ***** GX | 13-340               |
| \$DIGIT   | = 000024   | #10-164              |
| \$DNUMB   | = 000014   | #10-164              |
| \$EOS     | = 000012   | #10-164              |
| \$ERRD1   | 000274 R   | #8-123 12-282        |
| \$ERRD2   | 000332 R   | #8-124 13-361        |
| \$ERRP1   | 000014 R   | #8-117 11-238        |
| \$ERRP2   | 000052 R   | #8-118 11-239        |
| \$ERRP3   | 000100 R   | #8-119 11-255        |
| \$ERRP4   | 000130 R   | #8-120 11-267        |
| \$ERRP5   | 000154 R   | #8-121 13-362        |
| \$ERRP6   | 000226 R   | #8-122 13-363        |
| \$ERRIT   | = ***** GX | 9-158                |
| \$EXIT    | = 000000   | #10-164              |
| \$FAIL    | = 177777   | #10-164              |
| \$GPRM    | = *****    | 10-164               |
| \$HEADR   | = ***** GX | 13-335               |
| \$LAMDA   | = 000000   | #10-164              |
| \$NUMBR   | = 000002   | #10-164              |
| \$QPSN    | 000000 RG  | #9-143               |
| \$RAD50   | = 000016   | #10-164              |
| \$RONLY   | = *****    | 10-164 10-164        |
| \$STRNG   | = 000004   | #10-164              |
| \$SUBXP   | = 000010   | #10-164              |
| \$XALOC   | = ***** GX | 13-333               |
| \$\$\$FLG | = 177777   | #10-164              |
| \$\$\$KEY | = 177777   | #10-164              |
| .PNUMB    | = ***** GX | 11-246 11-262 12-289 |
| .PNUMH    | = ***** GX | 11-244 11-260 12-287 |
| .PSTCN    | = ***** GX | 11-217               |
| .PSTPT    | = ***** GX | 11-227               |
| .TPARS    | = ***** GX | 9-149                |

```

342
343
344      ; MAXIMUM BLOCK SIZE
345
346      PVPSZ: CALL    CHKBLK      ; CHECK FOR LEGAL BLOCK SIZE VALUE
347      BCS     101$           ; BR IF ERROR
348      MOV     R0,PVCPSZ      ; SAVE MAXIMUM BLOCK SIZE
349      RETURN
350
351      ; ERRORS
352
353      101$: MSG$R P8          ; ILLEGAL MAXIMUM BLOCK SIZE
354
355      ; WINDOW SIZE
356
357      PVWSZ: CALL    CHKWND      ; CHECK FOR LEGAL WINDOW SIZE VALUE
358      BCS     101$           ; BR IF ILLEGAL VALUE
359      MOV     R0,PVCWSZ      ; SAVE WINDOW SIZE
360      RETURN
361
362      ; ERRORS
363
364      101$: MSG$R 00          ; ILLEGAL WINDOW SIZE
365
366      ; PROCESS OWNER
367
368      PVOWN: CMP     #3,,PSTCN   ; LEGAL PROCESS OWNER?
369      BLD     101$           ; BR IF NO
370      MOV     ,PSTPT,R0        ; POINT TO STRING TO CONVERT TO RAD50
371      CLR     R1              ; PERIOD IS NOT A VALID TERMINATOR
372      CALL    $CAT5           ; CONVERT TO RAD50
373      MOV     R0,PVCOWN      ; SAVE PROCESS OWNER NAME
374      RETURN
375
376      ; ERRORS
377
378      101$: MSG$R N1          ; ILLEGAL PROCESS OWNER NAME
379
380      ; FLAGS BYTE
381
382      PVFG: TST     ,PNUMH      ; LEGAL VALUE?
383      BNE     101$           ; BR IF NO
384      TSTB    ,PNUMB+1        ; LEGAL VALUE?
385      BNE     101$           ; BR IF NO
386      MOVB    ,PNUMB,PVCFLG   ; SAVE FLAGS BYTE
387      PVEND   ; CREATE PVC NAME BLOCK AND XCB
388      RETURN
389
390      ; ERRORS
391
392      101$: MSG$R N2          ; ILLEGAL FLAGS BYTE VALUE
393
394
395

```

CFPPVC      CREATED BY    MACRO    ON 29-JUN-85 AT 00:19      PAGE 3      I 10

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL    | VALUE      | REFERENCES                                       |
|-----------|------------|--|
| \$DIGIT   | = 000024   | #10-200  |
| \$DNUMB   | = 000014   | #10-200  |
| \$EOS     | = 000012   | #10-200  |
| \$ERRN0   | 000040 R   | #8-156 11-312 13-444 20-695                      |
| \$ERRN1   | 000100 R   | #8-157 12-380                                    |
| \$ERRN2   | 000134 R   | #8-158 12-395                                    |
| \$ERRN3   | 000166 R   | #8-159 13-445                                    |
| \$ERR00   | = ***** GX | 12-365   |
| \$ERR05   | = ***** GX | 11-340   |
| \$ERR09   | = ***** GX | 11-328   |
| \$ERRP8   | = ***** GX | 12-353   |
| \$ERR1T   | = ***** GX | 9-194  |
| \$EXIT    | = 000000   | #10-200  |
| \$FAIL    | = 177777   | #10-200  |
| \$GPRM    | = *****    | 10-200   |
| \$HEADR   | = ***** GX | 13-409 13-427 13-432 15-514 20-683               |
| \$LAMDA   | = 000000   | #10-200  |
| \$NUMBR   | = 000002   | #10-200  |
| \$QPVC    | 000000 RG  | #9-178   |
| \$RAD50   | = 000016   | #10-200  |
| \$RONLY   | = *****    | 10-200 10-200                                    |
| \$STRNG   | = 000004   | #10-200  |
| \$SUBXP   | = 000010   | #10-200  |
| \$XALOC   | = ***** GX | 13-407 14-462                                    |
| \$\$\$FLG | = 177777   | #10-200  |
| \$\$\$KEY | = 177777   | #10-200  |
| .PNUMB    | = ***** GX | 11-319 11-335 12-387 12-389 20-700 20-702 20-705 |
| .PNUMH    | = ***** GX | 11-317 11-333 12-385 20-703                      |
| .PSTCN    | = ***** GX | 11-291 12-370                                    |
| .PSTPT    | = ***** GX | 11-299 12-372                                    |
| .TPARS    | = ***** GX | 9-185  |

```

278                                     .SBTTL  CHKRD - CHECK FOR UNIQUE RDT
279
280                                     ;+
281                                     :
282                                     :   CHKRD - CHECK FOR UNIQUE REMOTE DTE NAME
283                                     :
284                                     :   INPUTS:
285                                     :       RDTNAM - SPECIFIED RDT NAME
286                                     :
287                                     :   OUTPUTS:
288                                     :       CARRY CLEAR - REMOTE DTE NAME IS UNIQUE
289                                     :       CARRY SET - REMOTE DTE NAME IS NOT UNIQUE
290                                     :       R0 DESTROYED
291                                     :
292                                     :
293 000444 017700 000000G   CHKRD:  MOV    @PSIPT,R0      ; GET ADDRESS OF PSI HOME BLOCK
294 000450 062700 000004   ADD     #H$RDT,R0      ; POINT TO REMOTE DTE LISTHEAD
295
296 000454                                     SWSTK$  30$      ; ENTER SYSTEM STATE
297 000460                                     SAVMAP   ; SAVE CURRENT MAPPING
298 000464 011000   10$:  MOV     (R0),R0      ; GET NEXT BLOCK IN LIST
299 000466 001424   BEQ     25$      ; BR IF END OF LIST
300 000470 010046   MOV     R0,-(SP)      ; SET UNMAPPED ADDRESS FOR CONVERSION
301 000472                                     CALL    $CEACK      ; CONVERT TO MAPPED ADDRESS
302 000476 012600   MOV     (SP)+,R0      ; RETRIEVE MAPPED ADDRESS
303 000500 010001   MOV     R0,R1      ; GET ADDRESS OF REMOTE DTE BLOCK
304 000502 062701 000002   ADD     #R$NAM,R1      ; POINT TO REMOTE DTE NAME
305 000506 012702 000000'   MOV     #RDTNAM,R2      ; POINT TO SPECIFIED REMOTE DTE NAME
306 000512 012703 000006   MOV     #RDTMX,R3      ; GET LENGTH OF NAME
307 000516 122122   20$:  CMPB    (R1)+,(R2)+      ; IS THE NAME UNIQUE?
308 000520 001361   BNE     10$      ; BR IF YES
309 000522 005303   DEC     R3      ; MORE TO CHECK?
310 000524 003374   BGT     20$      ; BR IF YES
311 000526                                     RETC     R0      ; ELSE NOT UNIQUE - SET USER C-BIT
312 000540   25$:  RESMAP   ; RESTORE PREVIOUS MAPPING
313 000544   30$:  RETURN  ;
314
315 000001                                     .END

```

|                  |                    |                  |                  |                      |
|------------------|--------------------|------------------|------------------|----------------------|
| A\$CHK= 000000   | HF\$HOS= 000004    | H\$X29C 000040   | P\$WRD= 000000   | ZTIME = ***** GX     |
| A\$CPS= 000000   | HF\$XDF= 000020    | I\$RAR= 000000   | Q\$ACTC 000004   | \$ALL16= ***** GX    |
| A\$PRI= 000000   | H\$CUG 000010      | I\$RDN= 000000   | Q\$AUC 000006    | \$ALPHA= 000022      |
| A\$TRP= 000000   | H\$DST 000012      | K\$CNT= 177546   | Q\$CLEN= 000006  | \$ANY = 000020       |
| CERR = ***** GX  | H\$D29 000014      | K\$CSR= 177546   | Q\$CTIM 000000   | \$BLANK= 000006      |
| CFGBF = ***** GX | H\$FLG 000000      | K\$LDC= 000000   | Q\$ICRE 000014   | \$DIGIT= 000024      |
| CFGST = ***** GX | H\$GLEN 000104     | K\$TPS= 000074   | Q\$LEN 000016    | \$DNUMB= 000014      |
| CFLIN = ***** GX | H\$GLT 000044      | LD\$LP = 000000  | Q\$MXAC 000012   | \$EOS = 000012       |
| CTMR 000210R     | 002 H\$GNAM 000050 | L\$ASG= 000000   | Q\$MXVC 000002   | \$ERR1T= ***** GX    |
| C\$CKP= 000000   | H\$GNML= 000020    | L\$DRV= 000000   | Q\$TCLZ 000010   | \$ERR9A 000000R 002  |
| C\$ORE= 000400   | H\$GPT 000046      | L\$P11= 000001   | Q\$OPT= 000010   | \$ERR9B 000036R 002  |
| C\$RSH= 177564   | H\$HITS 000034     | L\$P11= 000000   | RTSPC = ***** GX | \$ERR9C 000076R 002  |
| D\$BUG= 177514   | H\$HLEN 000044     | MAXVC 000206R    | R\$DER= 000000   | \$ERR9D 000132R 002  |
| D\$ISK= 000000   | H\$LBDA 000070     | M\$CRB= 000124   | R\$K11= 000001   | \$EXIT = 000000      |
| D\$LL1= 000001   | H\$LBDN 000072     | M\$CRX= 000000   | R\$SND= 000000   | \$FAIL = 177777      |
| D\$SYNC= 000000  | H\$LDTE 000002     | M\$FCS= 000000   | R\$11M= 000000   | \$LAMDA= 000000      |
| D\$YNM= 000000   | H\$LEN 000042      | M\$MGE= 000000   | SYNERR= ***** GX | \$NUMBR= 000002      |
| END 000030R      | 003 H\$LOTS 000032 | M\$NET= 000000   | \$SWRG= 000000   | \$QX29 000000RG      |
| E\$XPR= 000000   | H\$NETW 000024     | M\$OVR= 000000   | \$YSZ= 007600    | \$RAD50= 000016      |
| FMT10 = ***** GX | H\$NML = 000006    | N\$ACC= 000001   | T\$KMG= 000000   | \$STRNG= 000004      |
| FMT8 = ***** GX  | H\$NPT 000022      | N\$BUE= 000001   | T\$MIN= 000000   | \$SUBXP= 000010      |
| FM.10 = 000000   | H\$PTB 000020      | N\$LDV= 000001   | V\$CTR= 001000   | \$FLG= 177777        |
| FM.8 = 000000    | H\$PVC 000006      | N\$MCP= 000001   | X\$DBT= 000000   | \$KEY= 000000        |
| F\$LVL= 000001   | H\$RDTE 000004     | N\$MLL= 000001   | X29CTM 000104R   | \$R= 000010          |
| G\$STPP= 000000  | H\$RNW 000042      | N\$MOV= 000010   | X29DF 000000R    | 003 \$STA= 000000    |
| G\$TSS= 000000   | H\$SVC 000036      | N\$NCT= 000001   | X29END 000130R   | \$TMP= 000000R 005   |
| G\$TTK= 000000   | H\$TRB 000016      | N\$PEM= 000001   | X29KW 000000RG   | 004 .PNUMB= ***** GX |
| G\$WRD= 000000   | H\$XAVL 000100     | PSIPT = ***** GX | X29MXC 000052R   | .PNUMB= ***** GX     |
| HF\$DLM= 000002  | H\$XBIA 000074     | P\$P45= 000000   | X29ST 000000RG   | 003 .TPARS= ***** GX |
| HF\$GWY= 000010  |                    |                  |                  |                      |

. ABS. 000104 000 (RW,I,GBL,ABS,OVR)  
000250 001 (RW,I,LCL,REL,CON)  
DATA 000212 002 (RW,D,LCL,REL,CON)  
\$STATE 000040 003 (RW,D,LCL,REL,CON)  
\$KIAB 000002 004 (RW,D,LCL,REL,CON)  
\$KSTR 000007 005 (RW,D,LCL,REL,CON)  
Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
Work file writes: 0  
Size of work file: 13360 Words ( 53 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:19.46  
SY:CFPX29.V2,[132,134]CFPX29/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFPX29



```

203                                     .SBTTL X3P$DF ACTION ROUTINES
204
205                                     ;
206                                     ; DEFAULT BLOCK SIZE
207
208 000060 BLKDEF: CALL CHKBLK ; CHECK FOR LEGAL BLOCK SIZE VALUE
209 000064 103403 BCS 101$ ; BR IF ILLEGAL VALUE
210 000066 010067 000000G MOV R0,BLKSZD ; STORE DEFAULT BLOCK SIZE
211 000072 RETURN
212
213                                     ;
214                                     ; ERRORS
215
216 000074 101$: MSG$R P7 ; ERROR - ILLEGAL DEFAULT BLOCK SIZE
217
218                                     ;
219                                     ; MAXIMUM BLOCK SIZE
220
221 000102 BLKMAX: CALL CHKBLK ; CHECK FOR LEGAL BLOCK SIZE VALUE
222 000106 103406 BCS 101$ ; BR IF ILLEGAL VALUE
223 000110 020067 000000G CMP R0,BLKSZD ; IS DEFAULT IN RANGE?
224 000114 103406 BLO 102$ ; BR IF NO
225 000116 010067 000000G MOV R0,BLKSZM ; STORE MAXIMUM BLOCK SIZE
226 000122 RETURN
227
228                                     ;
229                                     ; ERRORS
230
231 000124 101$: MSG$R P8 ; ERROR - ILLEGAL MAXIMUM BLOCK SIZE
232 000132 102$: MSG$R 01 ; ERROR - DEFAULT EXCEEDS MAXIMUM
233
234                                     ;
235                                     ; DEFAULT WINDOW SIZE
236
237 000140 WNDDEF: CALL CHKWND ; CHECK FOR LEGAL WINDOW SIZE VALUE
238 000144 103403 BCS 101$ ; BR IF ILLEGAL VALUE
239 000146 110067 000000G MOV R0,WNDSZD ; STORE DEFAULT WINDOW SIZE
240 000152 RETURN
241
242                                     ;
243                                     ; ERRORS
244
245 000154 101$: MSG$R P9 ; ERROR - ILLEGAL DEFAULT WINDOWSIZE
246
247                                     ;
248                                     ; MAXIMUM WINDOW SIZE
249
250 000162 WNDMAX: CALL CHKWND ; CHECK FOR LEGAL WINDOW SIZE VALUE
251 000166 103406 BCS 101$ ; BR IF ILLEGAL VALUE
252 000170 020067 000000G CMP R0,WNDSZD ; IS DEFAULT IN RANGE?
253 000174 103406 BLO 102$ ; BR IF NO
254 000176 010067 000000G MOV R0,WNDSZM ; STORE MAXIMUM WINDOW SIZE
255 000202 RETURN
256
257                                     ;
258                                     ; ERRORS
259
260 000204 101$: MSG$R 00 ; ERROR - ILLEGAL MAXIMUM WINDOW SIZE
261 000212 102$: MSG$R 02 ; ERROR - DEFAULT EXCEEDS MAXIMUM

```

```

229      ;+
230      ; PR.XFR - READ PROCESS TASK IMAGE INTO CORE
231      ; THIS ROUTINE IS CALLED TO READ THE PROCESS TASK IMAGE INTO CORE.
232      ;
233      ; INPUTS:
234      ; R1=START ADDRESS OF WHERE TO LOAD THE CETAB TASK IMAGE
235      ;
236      ; OUTPUTS:
237      ; C-BIT=SUCCESS/FAILURE
238      ; RO=DESTROYED
239      ; -
240      PR.XFR:
241      000272      MOV     R1,R0      ; COPY CETAB START ADDRESS
242      000272      CALL    $READ     ; READ CETAB INTO MEMORY
243      000274      BCS     101$      ; IF CS, FAILURE
244      000300      RETURN
245      000302
246
247      ; ERROR CONDITION
248
249      101$:
250      000304      MOV     $IOSB,$CLIOS ; COPY ERROR CODE
251      000312      MOV     #ERR34,ERROR ; PROCESS IMAGE READ FAILURE
252      000320      RETURN
253
254      000001      .END
255

```

LBNIO - LOGICAL BLOCK I/O ROUTI MACRO V05.03b Saturday 29-Jun-85 J 15 00:21  
Table of contents

|     |     |                                     |
|-----|-----|-------------------------------------|
| 5-  | 55  | MACRO DEFINITIONS                   |
| 6-  | 61  | LOCAL DATA                          |
| 7-  | 100 | \$CLOPE - OPEN A FILE               |
| 8-  | 120 | \$RLBL - READ LABEL BLOCK           |
| 9-  | 151 | \$READX - READ INTO EXTENDED MEMORY |
| 10- | 208 | \$READ - READ LOGICAL BLOCK         |

.TITLE LDUIC - FIND THE LOAD UIC FOR NTL  
.IDENT /V05.00/

COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

INSTALLED DISK/NETWORK UIC CALCULATION ROUTINE

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 27-FEB-78  
VERSION 2.0 RELEASE
- 2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0

```

140 .SBTTL CUG$DF STATE TABLES
141
142 ISTAT$ CUGST,CUGKW
143
144 ; CUG$DF STATE TABLES
145 ;
146 STATES CUGDF
147 TRANS %CUG$DF%,1,SYNERR
148
149 STATES
150 TRANS $STRNG,,CUNAM ; CLOSED USER GROUP NAME
151
152 STATES
153 TRANS <','>
154
155 STATES
156 TRANS $DIGIT,,CUNBST ; START OF CLOSED USER GROUP NUMBER
157
158 STATES CUNXT
159 TRANS $DIGIT,CUNXT,CUGDIG ; ANOTHER DIGIT
160 TRANS $LAMDA,,CUNBEN ; END OF CUG NUMBER
161
162 STATES
163 TRANS !END,$EXIT,CUEND
164 TRANS <','>
165
166 STATES
167 TRANS $NUMBR,,CUFG ; CLOSED USER GROUP FLAGS BYTE
168
169 ; CHECK FOR END OF SOURCE LINE
170 ;
171 STATES END
172 TRANS <','>,$EXIT
173 TRANS $EOS,$EXIT
174
175 STATES

```

55  
56  
57  
58  
59  
60

.SBTTL MACRO DEFINITIONS

;  
; LIBRARY MACROS  
;

.MCALL EMSG\$,NTLERS\$,SAVRG,RESRG,ASL\$,STAT\$,TRANS\$,ISTAT\$

## SYMBOL CROSS REFERENCE

CREF 04.00

| SYMBOL    | VALUE      | REFERENCES    |
|-----------|------------|---------------|
| \$ERRM0   | 000106 R   | #7-78 12-252  |
| \$ERRM2   | 000136 R   | #7-79 14-312  |
| \$ERRM3   | 000166 R   | #7-80 14-299  |
| \$ERRN2   | 000000 R   | #7-75 13-284  |
| \$ERRN8   | 000022 R   | #7-76 11-213  |
| \$ERRN9   | 000056 R   | #7-77 12-237  |
| \$ERR11   | = ***** GX | 8-118         |
| \$EXIT    | = 000000   | #9-125        |
| \$FAIL    | = 177777   | #9-125        |
| \$GPRM    | = *****    | 9-125         |
| \$LAMDA   | = 000000   | #9-125        |
| \$NUMBR   | = 000002   | #9-125        |
| \$QDSA    | 000000 RG  | #8-101        |
| \$QDSC    | 000006 RG  | #8-103        |
| \$RAD50   | = 000016   | #9-125        |
| \$RONLY   | = *****    | 9-125 9-125   |
| \$SAVRG   | = ***** GX | 11-218        |
| \$STRNG   | = 000004   | #9-125        |
| \$SUBXP   | = 000010   | #9-125        |
| \$\$\$FLG | = 177777   | #9-125        |
| \$\$\$KEY | = 177777   | #9-125        |
| .PNUMB    | = ***** GX | 12-259        |
| .PNUMH    | = ***** GX | 12-257        |
| .PSTCN    | = ***** GX | 13-272 14-315 |
| .PSTPT    | = ***** GX | 13-276 14-319 |
| .TPARS    | = ***** GX | 8-109         |

```

278                                     .SBITL DSTEND - END OF DESTINATION DESCRIPTOR BLOCK MACROS
279
280                                     ;
281                                     ; DSTEND - END OF DESTINATION DESCRIPTOR BLOCK MACROS
282
283 000664 126767 000000G 000000G DSTEND::CMPB DSCMCT,DSCVCT ; ARE CALL DATA MASK/VALUE SAME LENGTH?
284 000672 001044 BNE 104$ ; BR IF NO - ERROR
285 000674 017700 000000G MOV @PSIPT,R0 ; POINT TO PSI HOME BLOCK
286 000700 132767 000000G 000000G BITB #DS.X29,DSFLG ; IS THIS AN X29 DESCRIPTOR BLOCK?
287 000706 001003 BNE 10$ ; BR IF YES
288 000710 062700 000012 ADD #HSDST,R0 ; ELSE POINT TO X25 LISTHEAD
289 000714 000402 BR 20$ ; CONTINUE IN COMMON CODE
290 000716 062700 000014 10$: ADD #HSD29,R0 ; POINT TO X29 LISTHEAD
291 000722 20$: CALL CHKDSN ; CHECK UNIQUENESS OF DESTINATION NAME
292 000726 103420 BCS 101$ ; BR IF ERROR
293 000730 012701 000032 MOV #DSFLEN,R1 ; GET LENGTH OF FIXED FIELDS
294 000734 012700 000000G MOV #DSTVAR,R0 ; POINT TO VARIABLE FIELDS TABLE
295 000740 012002 30$: MOV (R0)+,R2 ; GET ADDRESS OF COUNT FIELD
296 000742 001406 BEQ 50$ ; BR IF END OF TABLE
297 000744 111202 MOVB (R2),R2 ; GET COUNT FIELD
298 000746 006202 ASR R2 ; GET BYTE COUNT (FROM PACKED COUNTS)
299 000750 005502 ADC R2 ;
300 000752 060201 ADD R2,R1 ; UPDATE DESTINATION DESCRIPTOR LENGTH
301 000754 005720 TST (R0)+ ; POINT PAST ADDRESS OF SPECIFIED FIELD
302 000756 000770 BR 30$ ; GET NEXT FIELD
303 000760 50$: CALL ALLDST ; ALLOCATE DESTINATION DESCRIPTOR BLOCK
304 000764 103404 BCS 103$ ; BR IF UNSUCCESSFUL ALLOCATION
305 000766 RETURN
306
307 ; ERRORS
308
309 000770 101$: MSG$R M7 ; DESTINATION NAME NOT UNIQUE
310 000776 103$: MSG$R M9 ; DESTINATION BLOCK ALLOCATION FAILURE
311 001004 104$: MSG$R GM ; ILLEGAL CALL DATA MASK/VALUE

```



```

55                                     .SBTTL  MACRO DEFINITIONS
56
57                                     ;
58                                     ; LIBRARY MACROS
59                                     ;
60                                     .MCALL  MSGSR$,SLTDF$,SAVRG,RESRG,DTEDF$,RETC
61                                     .MCALL  PHBDF$,NTLR$,ISTAT$,TRAN$,STATE$
62                                     ;
63                                     ; LIBRARY SYMBOLS
64                                     ;
65                                     DTEDF$          ; LOCAL DTE DESCRIPTOR BLOCK OFFSETS
66                                     PHBDF$          ; DEFINE PSI HOME BLOCK OFFSETS
67                                     SLTDF$          ; DEFINE SLT OFFSETS
68                                     ;
69                                     ; LOCAL MACRO DEFINITIONS
70                                     ;
71                                     .MACRO  SAVMAP
72                                     MOV     @KSAR5,-(SP)    ; SAVE APR 5
73                                     .ENDM
74
75                                     .MACRO  RESMAP
76                                     MOV     (SP)+,@KSAR5    ; RESTORE APR5
77                                     .ENDM

```

464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514

.SBTTL CHKDEV - CHECK VALIDITY OF LINE-ID

```

+
**--CHKDEV-CHECK VALIDITY OF LINE-ID
THIS ROUTINE MAKES SURE THAT THE DEVICE SPECIFIED IS KNOWN TO THE SYSTEM,
AND THAT THE FIELDS SPECIFIED APPLY.
INPUTS:
  LINNAM = DDM NAME OF LINE IN RAD50
  LINCTL = CONTROLLER NUMBER
  LINUNT = UNIT NUMBER
OUTPUTS:
  IF CC, LINE IS FOUND AND IS LEGAL
  R4 = SYSTEM LINE NUMBER
  ELSE, UNRECOGNIZED LINE
-
CHKDEV:
  SWSTK$ 110$                ;; ENTER SYSTEM STATE
  MAKE SURE THE PROCESS EXISTS, AND IS A DDM
  MOV LINNAM,R0                ;; GET DDM NAME TO LOOK FOR
  CALL FNDPDV                  ;; FIND IT'S PDV ADDRESS
  BCS 80$                       ;; IF CC, LINE NOT IN SYSTEM
  MAKE SURE THE LINE EXISTS
  MOV #-1,R4                    ;; CALCULATE SYSTEM LINE NUMBER
  MOV @SLINM,R0                 ;; GET NUMBER OF SYSTEM LINES
  MOV @SLIMA,R3                 ;; GET ADDRESS OF SYSTEM LINE INDEX TABLE
  10$: MOV (R3)+,R2              ;; GET ADDR OF SYSTEM LINE TABLE ENTRY
  INC R4                        ;; UPDATE SYSTEM LINE NUMBER
  CMPB L,DDM(R2),R1             ;; DOES THIS PROCESS MATCH ?
  BNE 20$                       ;; IF NE, NO - KEEP LOOKING
  CMPB L,CTL(R2),LINCTL         ;; DOES THIS CONTROLLER NUMBER MATCH ?
  BNE 20$                       ;; IF NE, NO - KEEP LOOKING
  CMPB L,UNT(R2),LINUNT         ;; DOES THIS UNIT NUMBER MATCH ?
  BEQ 30$                       ;; IF EQ, FOUND THE CORRECT SLN
  20$: DEC R0                    ;; LOOK THROUGH ALL SYSTEM LINES
  BNE 10$                       ;;
  BR 80$                        ;; IF HERE, LINE NOT FOUND
  30$: BIT #LF.MTP,L.FLG(R2)    ;; IS THIS A MULTI-DROP LINE?
  BNE 80$                       ;; IF NE, YES - ERROR
  BR 100$                      ;; ELSE VALID LINE ID
  80$: RETC R0                  ;; SET/CLEAR USER CARRY
  100$: MOV R4,R$R4(SP)         ;; SET USER R4 TO SLN
  110$: RETURN                  ;; AND RETURN TO USER MODE

```

```

LOCAL SYMBOLS

80                                     .SBTTL  LOCAL SYMBOLS
81
82      000006                       PNAMMX = 6           ; MAXIMUM CHARACTERS IN PSI NETWORK NAME
83      000400                       PORTMX = 256.         ; MAXIMUM NUMBER OF PORTS
84      000040                       SPACE = 40           ; ASCII SPACE
85
86                                     :
87      : SAVED REGISTER OFFSETS FOR SWSTK$
88      :
89      000002                       R$R0 = 2             ; R0
90      000004                       R$R1 = 4             ; R1
91      000006                       R$R2 = 6             ; R2
92      000010                       R$R3 = 10            ; R3
93      000012                       R$R4 = 12            ; R4
94      000014                       R$R5 = 14            ; R5

```

CFPPSN CREATED BY MACRO ON 29-JUN-85 AT 00:18 PAGE 3 J 8

MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES |         |         |         |         |         |         |         |         |         |
|------------|------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| CALL       | 9-149      | 12-273  | 13-303  | 13-313  | 13-332  | 13-333  | 13-340  |         |         |         |
| DBGTP\$    | #10-164    | #10-183 | #10-187 | #10-191 | #10-195 | #10-199 | #10-206 |         |         |         |
| EMSG\$R    | #5-60      | 9-158   | 11-238  | 11-239  | 11-255  | 11-267  | 12-282  | 13-361  | 13-362  | 13-363  |
| ISTAT\$    | #5-61      | 10-164  |         |         |         |         |         |         |         |         |
| MTRAN\$    | #10-164    |         |         |         |         |         |         |         |         |         |
| NTLER\$    | #5-60      | 8-117   | 8-118   | 8-119   | 8-120   | 8-121   | 8-122   | 8-123   | 8-124   |         |
| PHBDF\$    | #5-60      | 5-66    |         |         |         |         |         |         |         |         |
| RESMAP     | #5-75      | 13-346  |         |         |         |         |         |         |         |         |
| RESRG      | #5-60      | 11-226  | 13-329  |         |         |         |         |         |         |         |
| RETC       | #5-60      | 13-335  |         |         |         |         |         |         |         |         |
| RETURN     | 9-153      | 11-234  | 11-251  | 11-263  | 12-278  | 12-296  | 13-347  | 13-357  |         |         |
| SAVMAP     | #5-71      | 13-338  |         |         |         |         |         |         |         |         |
| SAVRG      | #5-60      | 11-222  | 13-315  |         |         |         |         |         |         |         |
| STATE\$    | #5-61      | 10-169  | #10-172 | #10-175 | #10-178 | #10-181 | #10-185 | #10-189 | #10-193 | #10-197 |
|            | #10-201    | #10-204 | #10-208 |         |         |         |         |         |         |         |
| SWSTK\$    | 13-332     |         |         |         |         |         |         |         |         |         |
| TRANS      | #5-61      | #10-170 | #10-173 | #10-176 | #10-179 | #10-182 | #10-183 | #10-186 | #10-187 | #10-190 |
|            | #10-191    | #10-194 | #10-195 | #10-198 | #10-199 | #10-202 | #10-205 | #10-206 |         |         |

```

397
398
399          ; END OF PVC$DF MACRO
400
401 000404          PVEND: CALL    CHKPVC          ; CHECK FOR VALID PVC ID
402 000410          BCS     101$          ; BR IF ERROR
403 000412 103505   MOV     #C$LEN,R1      ; GET LENGTH OF BLOCK TO ALLOCATE
404
405 000416          20$: SWSTK$ 120$          ; ENTER SYSTEM STATE
406 000422          SAVMAP          ; SAVE PREVIOUS MAPPING
407 000426          CALL    $XALOC        ; ALLOCATE A BLOCK FROM NETWORK POOL
408 000432 103006   BCC     60$          ; BR IF SUCCESSFUL ALLOCATION
409 000434          RETC    R0           ; ELSE SET USER C-BIT
410 000446          BR      110$         ; AND EXIT
411 000450 000461   MOV     R0,R5        ; SAVE UNMAPPED ADDRESS OF ALLOCATED BLOCK
412 000452 010005   MOV     R0,-(SP)     ; SET UP UNMAPPED ADDRESS FOR CONVERSION
413 000454          CALL    $CEACX       ; CONVERT TO APR5 MAPPED ADDRESS
414 000460          MOV     (SP)+,R0     ; RETRIEVE MAPPED ADDRESS
415 000462 010004   MOV     R0,R4        ; SAVE MAPPED ADDRESS
416 000464 062700   ADD     #C$NAM,R0   ; POINT TO CIRCUIT ID FIELD
417 000470 012702 000002 MOV     #PVNAMX,R2 ; GET LENGTH OF CIRCUIT ID
418 000474 012701 000006 MOV     #PVCNAM,R1 ; POINT TO SPECIFIED CIRCUIT ID
419 000500 112120   70$: MOVB    (R1)+,(R0)+ ; STORE CIRCUIT ID
420 000502 005302   DEC     R2          ; MORE TO STORE?
421 000504 003375   BGT     70$         ; BR IF YES
422 000506 116764 000023* 000015 MOVB    PVCFLG,C$FLG(R4) ; STORE FLAGS BYTE
423 000514 016764 000002* 000010 MOV     PVCLCN,C$LCN(R4) ; STORE LOGICAL CHANNEL NUMBER
424 000522 016764 000000G 000012 MOV     DTEDES,C$DTE(R4) ; STORE DTE DESCRIPTOR ADDRESS
425 000530          80$: CALL    FNDPRT      ; FIND FREE PORT ENTRY
426 000534 103006   BCC     90$         ; BR IF FOUND ONE
427 000536          RETC    R0          ; ELSE SET USER C-BIT
428 000550 000412   BR      100$        ; AND EXIT
429 000552 110164 000014   90$: MOVB    R1,C$PORT(R4) ; STORE PORT NUMBER
430 000556          CALL    ALLXCB       ; ALLOCATE X25 CIRCUIT BLOCK
431 000562 103005   BCC     100$        ; BR IF SUCCESS
432 000564          RETC    R0          ; ELSE SET USER C-BIT
433 000576 017700 000000G MOV     @PSIPT,R0 ; POINT TO PSI HOME BLOCK
434 000602 062700 000006 ADD     #H$PVC,R0 ; POINT TO PVC BLOCK LISTHEAD
435 000606          CALL    LNKEND       ; LINK CURRENT BLOCK (IN R5) TO LIST
436 000612          110$: RESMAP        ; RESTORE PREVIOUS MAPPING
437 000616          RETURN
438
439 000620 103404   120$: BCS     102$          ; BR IF ERROR
440 000622          RETURN
441
442          ; ERRORS
443
444 000624          101$: MSG$R NO          ; INVALID PVC ID
445 000632          102$: MSG$R N3        ; PVC/XCB/PORT/HASH ALLOCATION FAILED

```

CFPPVC CREATED BY MACRO ON 29-JUN-85 AT 00:19 PAGE 4 J 10  
 MACRO CROSS REFERENCE CREF 04.00

| MACRO NAME | REFERENCES  |
|------------|---|
| BIAS       | #5-86   |
| CALL       | 9-185 12-346 12-358 12-374 12-390 13-401 13-405 13-407 13-413 13-425            |
|            | 13-430 13-435 14-462 14-466 14-469 14-470 14-474 15-499 15-504 16-538           |
|            | 17-572 17-580 18-617 20-675 20-680 20-681                                       |
| DBGTP\$    | #10-200 #10-213 #10-229 #10-233 #10-237 #10-241 #10-245 #10-249 #10-253 #10-257 |
|            | #10-261 #10-265   |
| EMSG\$R    | #5-59 9-194 11-312 11-328 11-340 12-353 12-365 12-380 12-395 13-444             |
|            | 13-445 20-695   |
| ISTAT\$    | #5-60 10-200  |
| MAP        | #5-80   |
| MTRANS     | #10-200   |
| NTLER\$    | #5-60 8-156 8-157 8-158 8-159   |
| PHBDF\$    | #5-60 5-64  |
| PVCDF\$    | #5-59 5-65  |
| RESMAP     | #5-74 13-436 15-515 16-552 18-638   |
| RESRG      | #5-59   |
| RETC       | #5-59 13-409 13-427 13-432 15-514 20-683  |
| RETURN     | 9-189 11-308 11-324 11-336 12-349 12-361 12-376 12-391 13-437 13-440            |
|            | 14-479 15-516 16-553 17-597 18-639 19-659 20-688 20-691 20-707                  |
| SAVMAP     | #5-70 13-406 15-500 16-535 18-615   |
| SAVRG      | #5-59   |
| STATE\$    | #5-60 10-205 #10-208 #10-211 #10-215 #10-218 #10-221 #10-224 #10-227 #10-231    |
|            | #10-235 #10-239 #10-243 #10-247 #10-251 #10-255 #10-259 #10-263 #10-269 #10-272 |
|            | #10-275 #10-278 #10-281 #10-284   |
| SWSTK\$    | 13-405 15-499 20-680  |
| TRANS      | #5-60 #10-206 #10-209 #10-212 #10-213 #10-216 #10-219 #10-222 #10-225 #10-228   |
|            | #10-229 #10-232 #10-233 #10-236 #10-237 #10-240 #10-241 #10-244 #10-245 #10-248 |
|            | #10-249 #10-252 #10-253 #10-256 #10-257 #10-260 #10-261 #10-264 #10-265 #10-270 |
|            | #10-273 #10-276 #10-279 #10-282   |
| XCBD\$     | #5-59 5-66  |

```

A$$CHK= 000000      HF$XDF= 000020      K$$CNT= 177546      RDIDF= 000000R      003 X$$DBT= 000000
A$$CPS= 000000      H$CUG= 000010      K$$CSR= 177546      RDTE= 000006R      002 $ALPHA= 000022
A$$PRI= 000000      H$DST= 000012      K$$LDC= 000000      RDTMX= 000006      $ANY= 000020
A$$TRP= 000000      H$D29= 000014      K$$TPS= 000074      RDTE= 000024R      002 $BLANK= 000006
CERR= ***** GX   H$FLG= 000000      LNKEND= ***** GX  RDTE= 000166R      $CEACX= ***** GX
CFGBF= ***** GX   H$GLEN= 000104      L$ASG= 000000      RDTENO= 000166R      $DIGIT= 000024
CFGSZ= ***** GX   H$GLT= 000044      L$DRV= 000000      RDTE= 000020R      003 $DNUMB= 000014
CFLIN= ***** GX   H$GNAM= 000050      L$SP11= 000001      RDTST= 000025R      $EOS= 000012
CHKRDT= 000444R      H$GNML= 000020      L$SP11R= 000000      RDTE= 000130R      004 $ERRLO= 000036R      002
COUNT= ***** GX H$GPT= 000046      L$SRB= 000124      RDTE= 000017      $ERR1= 000066R      002
C$$CKP= 000000      H$HITS= 000034      M$CRX= 000000      RDTNAM= 000000R      002 $ERR2= 000122R      002
C$SORE= 000400      H$HLEN= 000044      M$FCR= 000000      RDTST= 000000R      003 $ERRIT= ***** GX
C$SRSH= 177564      H$LBDA= 000070      M$MGE= 000000      RTSPC= ***** GX  $EXIT= 000000
D$BBUG= 177514      H$LBDN= 000072      M$NET= 000000      RSDAL= 000013      $FAIL= 177777
D$BTK= 000000      H$LDTE= 000002      M$OVR= 000000      RSDTE= 000014      $HEADR= ***** GX
D$BL11= 000001      H$LEN= 000042      M$SACC= 000001      R$LEN= 000024      $LAMBDA= 000000
D$VNC= 000000      H$LOTS= 000032      M$SBUF= 000001      R$LNK= 000000      $NUMBR= 000002
D$VNM= 000000      H$NETW= 000024      M$SLDV= 000001      R$NAM= 000002      $QDRT= 000000RG
END= 000036R      003 H$NML= 000006      M$SMP= 000001      R$OWN= 000010      $RAD50= 000016
E$XPR= 000000      H$NPT= 000022      M$MOV= 000010      R$SND= 000000      $SAVRG= ***** GX
FMT10= ***** GX  H$PTB= 000020      M$MML= 000001      R$SND= 000000      $STRNG= 000004
FMT8= ***** GX   H$PVC= 000006      M$MOV= 000010      R$SND= 000000      $SUBXP= 000010
FM.10= 000000      H$RDTE= 000004      M$NCT= 000001      R$SND= 000000      $XALOC= ***** GX
FM.8= 000000      H$RNW= 000042      M$PEM= 000001      SPACE= 000040      $$$FLG= 177777
F$LV= 000001      H$SVC= 000036      PCKBC= ***** GX  STRNXT= ***** GX
G$TTP= 000000      H$TRB= 000016      PSIPT= ***** GX  SYNERR= ***** GX
G$TSS= 000000      H$XAVL= 000100      P$P45= 000000      S$WRG= 000000      $$$KEY= 000000
G$TIT= 000000      H$XBIA= 000074      P$WRD= 000000      S$YSZ= 007600      $$$R= 000010
G$WRD= 000000      H$X29C= 000040      Q$OPT= 000010      T$MIN= 000000      $$$STA= 000000
HF$DLM= 000002      I$RAR= 000000      RDEND= 000212R      V$CTR= 001000      $$$TMP= 000000R      005
HF$GWY= 000010      I$RDN= 000000      RDNAM= 000052R      X$DBT= 000000      .PSTCN= ***** GX
HF$HOS= 000004      K$ARS= ***** GX  Y$DBT= 000000      .PSTPT= ***** GX
. ABS. 000104 000 (RW,I,GBL,ABS,OVR)      .TPARS= ***** GX
          000546 001 (RW,I,LCL,REL,CON)
DATA 000176 002 (RW,D,LCL,REL,CON)
$STATE 000046 003 (RW,D,LCL,REL,CON)
$KTAB 000002 004 (RW,D,LCL,REL,CON)
$KSTR 000007 005 (RW,D,LCL,REL,CON)
Errors detected: 0

*** Assembler statistics

Work file reads: 0
Work file writes: 0
Size of work file: 13284 Words ( 52 Pages)
Size of core pool: 14440 Words ( 55 Pages)
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:22.57
SY:CFPRDT.V2,[132,134]CFPRDT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFPRDT

```

CFPX29      CREATED BY    MACRO    ON 29-JUN-85 AT 00:20      PAGE 1      J 12  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL   | VALUE      | REFERENCES                                   |
|----------|------------|--|
| CERR     | = ***** GX | 6-73      6-74      6-75      6-76           |
| CFGBF    | = ***** GX | 7-106  |
| CFGSZ    | = ***** GX | 7-105  |
| CFLIN    | = ***** GX | 6-73      6-74      6-75      6-76           |
| CTIMR    | = 000210 R | #6-85      *9-169      9-192                 |
| FMT10    | = ***** GX | 6-73      6-74      6-75                     |
| FMT8     | = ***** GX | 6-76   |
| FM.10    | = 000000   | #6-73      #6-74      #6-75                  |
| FM.8     | = 000000   | #6-76  |
| H\$X29C  | 000040     | 9-179      *9-198                            |
| MAXVC    | 000206 R   | #6-84      *9-159      9-191                 |
| PSIPT    | = ***** GX | 9-177      9-197                             |
| Q\$CTIM  | 000000     | *9-192                                       |
| Q\$LEN   | 000016     | 9-182      9-186                             |
| Q\$MXVC  | 000002     | *9-191                                       |
| Q\$TCLZ  | 000010     | *9-196                                       |
| RTSPC    | = ***** GX | 6-73      6-74      6-75      6-76           |
| SYNERR   | = ***** GX | *7-107      *7-110                           |
| \$B\$BAS | = *****    | 6-73      6-74      6-75      6-76      6-76 |
| X29CTM   | 000104 R   | #9-167                                       |
| X29DF    | 000000 R   | 7-102  |
| X29END   | 000130 R   | #9-177                                       |
| X29KW    | 000000 RG  | 7-104      #8-123                            |
| X29MXC   | 000052 R   | #9-155                                       |
| X29ST    | 000000 RG  | #8-123                                       |
| ZTIME    | = ***** GX | 9-194  |
| \$ALL16  | = ***** GX | 9-183  |
| \$ALPHA  | = 000022   | #8-123                                       |
| \$ANY    | = 000020   | #8-123                                       |
| \$BLANK  | = 000006   | #8-123                                       |
| \$DIGIT  | = 000024   | #8-123                                       |
| \$DNUMB  | = 000014   | #8-123                                       |
| \$EOS    | = 000012   | #8-123                                       |
| \$ERR1T  | = ***** GX | 7-117  |
| \$ERR9A  | 000000 R   | #6-73      9-201                             |
| \$ERR9B  | 000036 R   | #6-74      9-162                             |
| \$ERR9C  | 000076 R   | #6-75      9-172                             |
| \$ERR9D  | 000132 R   | #6-76      9-202                             |
| \$EXIT   | = 000000   | #8-123                                       |
| \$FAIL   | = 177777   | #8-123                                       |
| \$GPRM   | = *****    | 8-123  |
| \$LAMDA  | = 000000   | #8-123                                       |
| \$NUMBR  | = 000002   | #8-123                                       |
| \$QX29   | 000000 RG  | #7-102                                       |
| \$RAD50  | = 000016   | #8-123                                       |
| \$RONLY  | = *****    | 8-123      8-123                             |
| \$STRNG  | = 000004   | #8-123                                       |
| \$SUBXP  | = 000010   | #8-123                                       |
| \$B\$FLG | = 177777   | #8-123                                       |
| \$B\$KEY | = 177777   | #8-123                                       |
| .PNUMB   | = ***** GX | 9-159      9-169                             |
| .PNUMH   | = ***** GX | 9-157      9-167                             |



```

259                                     .SBTTL CHN$DF ACTION ROUTINES
260
261                                     : LOW END OF CHANNEL RANGE
262                                     :
263                                     :
264                                     : ENABL LSH
265 000220 005767 000000G CHLO: TST ,PNUMH ; IS THIS A LEGAL VALUE?
266 000224 001106 BNE 101$ ; BR IF NO
267 000226 016700 000000G MOV ,PNUMB,R0 ; GET SPECIFIED CHANNEL RANGE
268 000232 003503 BLE 101$ ; BR IF ILLEGAL VALUE
269 000234 022700 000000G CMP #CHNLMX,R0 ; LEGAL VALUE?
270 000240 103500 BLO 101$ ; BR IF NO
271 000242 010067 000000' MOV R0,CHNLLO ; SAVE LOW END VALUE
272 000246 010067 000002' MOV R0,CHNLHI ; THIS IS DEFAULT IF HIGH END VALUE OMITTED
273 000252 RETURN
274
275                                     : HIGH END OF CHANNEL RANGE
276                                     :
277                                     :
278 000254 005767 000000G CHHI: TST ,PNUMH ; IS THIS A LEGAL VALUE?
279 000260 001070 BNE 101$ ; BR IF NO
280 000262 016700 000000G MOV ,PNUMB,R0 ; GET SPECIFIED CHANNEL RANGE
281 000266 003465 BLE 101$ ; BR IF ILLEGAL VALUE
282 000270 022700 000000G CMP #CHNLMX,R0 ; LEGAL VALUE?
283 000274 103462 BLO 101$ ; BR IF NO
284 000276 010067 000002' MOV R0,CHNLHI ; SAVE HIGH END VALUE
285 000302 012701 000006 CHEND: MOV #K$LEN,R1 ; GET NUMBER OF BYTES TO ALLOCATE
286
287 000306 SWSTK$ 40$ ; ENTER SYSTEM STATE
288 000312 CALL @ALOCB ; ALLOCATE CHANNEL LIST FROM EXEC POOL
289 000316 103006 BCC 10$ ; BR IF SUCCESSFUL ALLOCATION
290 000320 RETC R0 ; SET USER C-BIT
291 000332 000440 BR 35$ ; AND EXIT
292 000334 016760 000000' 000002 10$: MOV CHNLLO,K$LOC(R0) ; STORE LOW END OF CHANNEL RANGE
293 000342 016760 000002' 000004 MOV CHNLHI,K$HIC(R0) ; STORE HIGH END OF CHANNEL RANGE
294 000350 SAVMAP ; SAVE CURRENT MAPPING
295 000354 016746 000000G MOV DTEDES,-(SP) ; GET DTE DESCRIPTOR ADDRESS
296 000360 $CEACX ; CONVERT TO MAPPED ADDRESS
297 000364 011601 MOV (SP),R1 ; RETRIEVE MAPPED ADDRESS
298
299 000366 062701 000034 20$: ADD #L$CHLS,R1 ; POINT TO OUTGOING CHANNELS LIST
300 000372 010102 MOV R1,R2 ; SAVE ADDRESS OF PREVIOUS BLOCK
301 000374 011101 MOV (R1),R1 ; GET NEXT BLOCK IN LIST
302 000376 001375 BNE 20$ ; BR IF NOT END OF LIST
303 000400 005010 CLR (R0) ; INITIALIZE LINK WORD
304 000402 010012 MOV R0,(R2) ; ADD BLOCK TO CHANNELS LIST
305
306 000404 016701 000002' MOV CHNLHI,R1 ; GET HIGH END OF CHANNELS
307 000410 166701 000000' SUB CHNLLO,R1 ; COMPUTE HIGH MINUS LOW
308 000414 002001 BGE 30$ ; IF GE, RESULT CORRECT
309 000416 005401 NEG R1 ; ELSE, GET ACTUAL DIFFERENCE
310 000420 005201 30$: INC R1 ; GET NUMBER OF CHANNELS
311 000422 062716 000036 ADD #L$MCHN,(SP) ; POINT TO MAX CHANNELS CELL
312 000426 060136 ADD R1,@(SP)+ ; ADJUST TOTAL CHANNEL COUNT
313 000430 RESMAP ; RESTORE PREVIOUS MAPPING
314 000434 35$: RETURN
315

```

```

Symbol table
A$$CHK= 000000      FE.X25= 004000      KINAR7= 172356      MPAR = 172100      SWR = 177570
A$$CPS= 000000      F$$LVL= 000001      KISAR0= 172340      MPCSR= 177746      S$$WRG= 000000
A$$PRI= 000000      F2.ACN= 000020      KISAR5= 172352      M$$CRB= 000124      S$$YSZ= 007600
A$$TRP= 000000      F2.AHR= 010000      KISAR6= 172354      M$$CRX= 000000      TPS = 177564
CMODE = 140000      F2.DAS= 000001      KISAR7= 172356      M$$FCS= 000000      T$$ACP= 020000
C$$CKP= 000000      F2.DPR= 000400      KISDR0= 172300      M$$MGE= 000000      T$$CHK= 000100
C$$ORE= 000400      F2.EVT= 000010      KISDR6= 172314      M$$NET= 000000      T$$CMP= 000200
C$$RSH= 177564      F2.GGF= 002000      KISDR7= 172316      M$$OVR= 000000      T$$IDP= 000020
DSKBUF= 000000R      003 F2.IRR= 001000      K$$CNT= 177546      N$$ACC= 000001      T$$NEW= 000001
D$$BUG= 177514      F2.LIB= 000002      K$$CSR= 177546      N$$BUF= 000001      T$$NHD= 040000
D$$ISK= 000000      F2.MP = 000004      K$$LDC= 000000      N$$LDV= 000001      T$$NSD= 002000
D$$L11= 000001      F2.PDL= 000100      K$$TPS= 000074      N$$MCP= 000001      T$$NXH= 000002
D$$YNC= 000000      F2.RAS= 004000      LB$REV= 000400      N$$MLL= 000001      T$$PIC= 100000
D$$YNM= 000000      F2.RBN= 020000      LD$ACC= 100000      N$$MOV= 000010      T$$PMD= 010000
ERROR 000000R      002 F2.SDW= 000040      LD$CLS= 020000      N$$NCT= 000001      T$$PRV= 000400
ERR10 000012      F2.STP= 100000      LD$LP = 000000      N$$PEM= 000001      T$$RES= 000040
ERR12 000014      F2.SWP= 040000      LD$REL= 000004      PIRQ = 177772      T$$SLV= 004000
ERR14 000016      F2.WND= 000200      LD$RSV= 040000      PMODE = 030000      T$$SUP= 000010
ERR16 000020      F3.AST= 000200      LD$SUP= 000010      PR.LBL 000076R      T$$XHR= 000004
ERR18 000022      F3.CL1= 001000      LB$ASG 001000      PR.XFR 000272R      T$$KMG= 000000
ERR2 000002      F3.CRA= 000001      LB$BLK 000360      PR0 = 000000      T$$MIN= 000000
ERR20 000024      F3.EIS= 000004      LB$BDAT 000032      PR1 = 000040      T2$CL1= 000001
ERR22 000026      F3.NWK= 000002      LB$EXT 000352      PR4 = 000200      T2$FMP= 000002
ERR24 000030      F3.PMN= 004000      LB$FLG 000030      PR5 = 000240      UBMPR = 170200
ERR26 000032      F3.PRO= 000040      LB$FL2 000774      PR6 = 000300      UDSAR0= 177660
ERR28 000034      F3.RLK= 020000      LB$HGV 000512      PR7 = 000340      UDSAR0= 177620
ERR30 000036      F3.SHF= 040000      LB$HRB 000356      PS = 177776      UISAR0= 177640
ERR32 000040      F3.STM= 000010      LB$LDZ 000016      P$$P45= 000000      UISAR4= 177650
ERR34 000042      F3.TCM= 002000      LB$LIB 000040      P$$WRD= 000000      UISAR5= 177652
ERR36 000044      F3.UDS= 000020      LB$LR1 000776      Q$$DPT= 000010      UISAR6= 177654
ERR4 000004      F3.WAT= 010000      LB$LUN 000362      R$LDAT 000026      UISAR7= 177656
ERR6 000006      F3.XHR= 000100      LB$MXV 000014      R$FLG 000024      UISAR0= 177600
ERR8 000010      F3.11S= 000400      LB$MXZ 000020      R$LHG 000006      UISDR4= 177610
E$$XPR= 000000      G$$TPP= 000000      LB$OFF 000022      R$LLDZ 000012      UISDR5= 177612
FE.CAL= 000040      G$$TSS= 000000      LB$PAR 000004      R$LMXV 000010      UISDR6= 177614
FE.CEX= 020000      G$$TTK= 000000      LB$PR1 000346      R$LMXZ 000014      UISDR7= 177616
FE.DRV= 000010      G$$WRD= 000000      LB$ROB 000364      R$LNAM 000000      V$$CTR= 001000
FE.DYM= 010000      HF.CIS= 000200      LB$ROL 000366      R$LOFF 000016      X$$DBT= 000000
FE.EXP= 000200      HF.EIS= 000002      LB$SA 000010      R$LSA 000004      $ALL16= ***** GX
FE.EXT= 000001      HF.FPP= 100000      LB$SEG 000026      R$LSEG 000022      $CLDEQ= ***** GX
FE.EXV= 000004      HF.QB = 000004      LB$SGL 000354      R$LSIZ 000034      $CLLOS= ***** GX
FE.FDT= 002000      HF.UBM= 000001      LB$SYS 000025      R$LWND 000020      $CLOPE= ***** GX
FE.LS1= 000400      I$$RAR= 000000      LB$TSK 000000      R$$DER= 000000      $CTBIO 000000RG
FE.MUP= 000002      I$$RDN= 000000      LB$WND 000024      R$$K11= 000001      $IOSB = ***** GX
FE.MX1= 040000      KDSAR0= 172360      LB$XFR 000350      R$$SND= 000000      $LBXL = 000340
FE.NLG= 100000      KDSDR0= 172320      L$$ASG= 000000      R$$11M= 000000      $LLEN = ***** GX
FE.OFF= 001000      KINAR0= 172340      L$$DRV= 000000      SISR0= 172200      $READ = ***** GX
FE.PKT= 000100      KINAR5= 172352      L$$P11= 000001      SR0 = 177572      $RLBL = ***** GX
FE.PLA= 000020      KINAR6= 172354      L$$11R= 000000      SR3 = 172516

. ABS. 001000 000 (RW,I,GBL,ABS,DVR)
000322 001 (RW,I,LCL,REL,CON)
DATA 000002 002 (RW,D,LCL,REL,CON)
.BUF 000002 003 (RW,D,LCL,REL,CON)
Errors detected: 0

*** Assembler statistics

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

.TITLE LBNIO - LOGICAL BLOCK I/O ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

LOGICAL BLOCK I/O ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 27-FEB-78  
VERSION 2.0 RELEASE
- 2.00 14-DEC-79  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0

```

55          .SBTTL  LOCAL DATA
56
57
58          ;
59          ; DEFINE TCB, UCB, AND DCB OFFSETS FOR RSX11-M
60          ;
61          .IF     DF,R$$11M          ;RSX11-M ONLY
62          .MCALL  TCBDF$,UCBDF$,DCBDF$
63
64          TCBDF$
65          UCBDF$
66          DCBDF$
67
68          .ENDC
69
70          ;
71          ; BUFFER USED TO BUILD NEW 'DV:[UIC]' STRING
72          ;
73          .PSECT  DATA,D          ;START OF LOCAL DATA PSECT
74          UICBF: .BLKW  10          ;ROOM FOR 17 CHARS (OCTAL)
75          .PSECT          ;START OF INSTRUCTION PSECT

```

```

177                                     .SBTTL  CUG$DF ACTION ROUTINES
178
179                                     ;
180                                     ; CUG NAME
181
182 000052 016700 000000G      CUNAM:  MOV    .PSTCN,R0      ; GET NUMBER OF CHARACTERS IN NAME
183 000056 020027 000000G      CMP      R0,#CUNMMX      ; LEGAL NAME?
184 000062 101022              BHI      101$             ; BR IF NO
185 000064 012701 000000'      MOV      #CUGNAM,R1      ; POINT TO CUG NAME
186 000070 012702 000000G      MOV      #CUNMMX,R2      ; GET LENGTH OF NAME
187 000074 112721 000040      10$:  MOVB   #SPACE,(R1)+  ; INITIALIZE TO SPACES
188 000100 005302              DEC      R2               ; MORE TO INITIALIZE?
189 000102 003374              BGT      10$              ; BR IF YES
190 000104 012701 000000'      MOV      #CUGNAM,R1      ; POINT TO CUG NAME
191 000110 016702 000000G      MOV      .PSTPT,R2      ; POINT TO SPECIFIED NAME
192 000114 112221              20$:  MOVB   (R2)+,(R1)+  ; STORE CUG NAME
193 000116 005300              DEC      R0               ; MORE TO STORE?
194 000120 003375              BGT      20$              ; BR IF YES
195 000122 105067 000014'      CLRB    CUGFLG          ; SET UP DEFAULT FOR FLAGS BYTE
196 000126
197                                     ;
198                                     ; ERRORS
199
200 000130      101$:  MSG$R  N2                          ; ILLLEGAL CUG NAME

```

LOCAL SYMBOL DEFINITIONS

62  
63  
64  
65  
66  
67

.SBTTL LOCAL SYMBOL DEFINITIONS

;  
; LOCAL SYMBOLS  
;

000040

DSCMX = 32.

; MAXIMUM NUMBER OF DIGITS FOR CALL DATA

CFPDSA      CREATED BY    MACRO    ON 29-JUN-85 AT 00:16      PAGE 3      K 3

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES |         |         |         |         |         |        |        |        |         |
|------------|------------|---------|---------|---------|---------|---------|--------|--------|--------|---------|
| ASL\$      | #5-60      | 14-334  |         |         |         |         |        |        |        |         |
| CALL       | 8-109      | 11-200  | 11-208  | 11-222  | 12-229  | 12-243  | 14-293 | 14-306 |        |         |
| DBGTP\$    | #9-125     | #9-134  | #9-135  | #9-136  | #9-140  | #9-141  | #9-142 | #9-146 | #9-150 | #9-154  |
|            | #9-158     | #10-173 | #10-183 |         |         |         |        |        |        |         |
| EMSG\$R    | #5-60      | 8-118   | 11-213  | 12-237  | 12-252  | 13-284  | 14-299 | 14-312 |        |         |
| ISTAT\$    | #5-60      | 9-125   |         |         |         |         |        |        |        |         |
| MTRANS     | #9-125     |         |         |         |         |         |        |        |        |         |
| NTLR\$     | #5-60      | 7-75    | 7-76    | 7-77    | 7-78    | 7-79    | 7-80   |        |        |         |
| RESRG      | #5-60      | 14-340  |         |         |         |         |        |        |        |         |
| RETURN     | 8-113      | 11-201  | 11-209  | 11-224  | 12-233  | 12-248  | 12-266 | 13-280 | 14-295 | 14-308  |
|            | 14-341     |         |         |         |         |         |        |        |        |         |
| SAVRG      | #5-60      | 14-314  |         |         |         |         |        |        |        |         |
| STATE\$    | #5-60      | 9-129   | #9-132  | #9-138  | #9-144  | #9-148  | #9-152 | #9-156 | #9-160 | 10-168  |
|            | #10-171    | #10-175 | #10-178 | #10-181 | #10-185 |         |        |        |        |         |
| TRANS      | #5-60      | #9-130  | #9-133  | #9-134  | #9-135  | #9-136  | #9-139 | #9-140 | #9-141 | #9-142  |
|            | #9-145     | #9-146  | #9-149  | #9-150  | #9-153  | #9-154  | #9-157 | #9-158 | #9-161 | #10-169 |
|            | #10-172    | #10-173 | #10-176 | #10-179 | #10-182 | #10-183 |        |        |        |         |

```

313 .SBTTL CHKDSN - CHECK DESTINATION NAME
314
315 ;+
316 CHKDSN - CHECK FOR UNIQUENESS OF DESTINATION NAME
317
318 INPUTS:
319 R0 - ADDRESS OF DESTINATION DESCRIPTOR LISTHEAD
320
321 OUTPUTS:
322 CARRY CLEAR - DESTINATION NAME IS UNIQUE
323 CARRY SET - DESTINATION NAME IS NOT UNIQUE
324
325
326
327 CHKDSN: SWSTK$ 40$          ;; ENTER SYSTEM STATE
328          SAVMAP          ;; SAVE CURRENT MAPPING
329 10$: MOV (R0),R0          ;; GET ADDRESS OF NEXT BLOCK
330      BEQ 30$             ;; BR IF END OF LIST
331      MOV R0, -(SP)        ;; SET UNMAPPED ADDRESS FOR CONVERSION
332      CALL $CEACX         ;; CONVERT TO MAPPED ADDRESS
333      MOV R0,R3           ;; RETRIEVE MAPPED ADDRESS
334      MOV R0,R3           ;; GET MAPPED ADDRESS
335      ADD #DSNAM,R3       ;; POINT TO DESTINATION NAME
336      MOV #DSTNAM,R1      ;; POINT TO SPECIFIED DESTINATION NAME
337      MOV #DSNAMX,R2      ;; GET LENGTH OF NAME
338 20$: CMPB (R3)+,(R1)+    ;; IS NAME UNIQUE?
339      BNE 10$             ;; BR IF YES
340      DEC R2              ;; MORE TO CHECK?
341      BGT 20$             ;; BR IF YES
342      RETC                ;; ELSE SET USER C-BIT
343 30$: RESMAP             ;; RESTORE PREVIOUS MAPPING
344 40$: RETURN
  
```

|     |        |                |
|-----|--------|----------------|
| 327 | 001012 |                |
| 328 | 001016 |                |
| 329 | 001022 | 011000         |
| 330 | 001024 | 001424         |
| 331 | 001026 | 010046         |
| 332 | 001030 |                |
| 333 | 001034 | 012600         |
| 334 | 001036 | 010003         |
| 335 | 001040 | 062703 000004  |
| 336 | 001044 | 012701 000000G |
| 337 | 001050 | 012702 000006  |
| 338 | 001054 | 122321         |
| 339 | 001056 | 001361         |
| 340 | 001060 | 005302         |
| 341 | 001062 | 003374         |
| 342 | 001064 |                |
| 343 | 001076 |                |
| 344 | 001102 |                |



79  
 80  
 81  
 82  
 83  
 84  
 85  
 86  
 87  
 88  
 89  
 90  
 91  
 92  
 93  
 94  
 95  
 96  
 97

000040  
 000040  
 001000  
 000020  
  
  
 000002  
 000004  
 000006  
 000010  
 000012  
 000014

.SBTTL LOCAL SYMBOL DEFINITIONS

;; LOCAL SYMBOL DEFINITIONS  
 ;

SPACE = 40 ; ASCII SPACE  
 HSHMN = 32 ; MINIMUM VALUE FOR HASH TABLE SIZE  
 HSHMX = 512 ; MAXIMUM VALUE FOR HASH TABLE SIZE  
 NUMDTE = 16 ; MAXIMUM DTES ALLOWED

;; SAVED REGISTER OFFSETS ON STACK FOR SWSTK\$  
 ;

R\$R0 = 2 ; SAVED R0  
 R\$R1 = 4 ; SAVED R1  
 R\$R2 = 6 ; SAVED R2  
 R\$R3 = 10 ; SAVED R3  
 R\$R4 = 12 ; SAVED R4  
 R\$R5 = 14 ; SAVED R5

```

516                                     .SBTTL BIASX - SET APR6 BIAS CORRECTLY
517
518                                     :+
519                                     :
520                                     : BIASX - SET APR6 BIAS CORRECTLY
521                                     :
522                                     : INPUTS:
523                                     :   R0 = ADDRESS (POOL OR APR5)
524                                     :
525                                     : OUTPUTS:
526                                     :   R0 = ADDRESS (POOL OR APR6)
527                                     :
528                                     :-
529
530 BIASX:                                CMP     R0,#120000          ;; ADDRESS IN APR5 ?
531 001206 020027 120000                BLO     10$                ;; BR IF NO (ASSUME POOL)
532 001212 103404                      BIC     #160000,R0          ;; CLEAR ALL APR BITS
533 001214 042700 160000                BIS     #140000,R0          ;; AND BIAS FOR APR6
534 001220 052700 140000                10$:  RETURN                ;; RETURN
535 001224
536
537 000001                                .END

```

```

LOCAL DATA
96          .SBTTL  LOCAL DATA
97
98          :
99          : LOCAL DATA
100         :
101
102         .PSECT  DATA,D
103
104         .NLIST  BEX
105
106         :
107         : LOCAL DATA FOR PSN$DF MACRO
108         PSNNAM: .BLKB 6          : PSN NETWORK NAME FROM PSN$DF MACRO
109         PSNFLG: .BLKW 1          : FLAGS WORD FROM PSN$DF MACRO
110         PSNLO:  .BLKW 1          : LOW END OF SUBADDRESS RANGE
111         PSNHI:  .BLKW 1          : HIGH END OF SUBADDRESS RANGE

```

\*\*FILE\*\*ID\*\*CFPPVC

```

CCCCCCCC FFFFFFFF PPPPPPP PPPPPPP VV VV CCCCCCCC
CCCCCCCC FFFFFFFF PPPPPPP PPPPPPP VV VV CCCCCCCC
CC FF PP PP PP PP VV VV CC
CC FF PP PP PP PP VV VV CC
CC FF PP PP PP PP VV VV CC
CC FFFF PP PPPPPPP PPPPPPP VV VV CC
CC FFFF PP PPPPPPP PPPPPPP VV VV CC
CC FF PP PP PP PP VV VV CC
CC FF PP PP PP PP VV VV CC
CC FF PP PP PP PP VV VV CC
CC FF PP PP PP PP VV VV CC
CCCCCCCC FF PP PP PP PP VV VV CCCCCCCC
CCCCCCCC FF PP PP PP PP VV VV CCCCCCCC

```

```

....
....
....
....

```

```

LL SSSSSSS TTTTTTTTT
LL SSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSS TT
LL SSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LLLLLLLLL SSSSSSS TT
LLLLLLLLL SSSSSSS TT

```

```

ALLXCB - ALLOCATE X25 CIRCUIT BLOCK

447                                     .SBTTL ALLXCB - ALLOCATE X25 CIRCUIT BLOCK
448
449
450
451                                     +
452                                     : ALLXCB - ALLOCATE AND SET UP X25 CIRCUIT BLOCK
453                                     :
454                                     : INPUTS:
455                                     :   PRTADD - ADDRESS OF ENTRY IN PORT TABLE
456                                     :
457                                     : OUTPUTS:
458                                     :   R3 - MAPPED ADDRESS OF X25 CIRCUIT BLOCK (APR 5 MAPPED TO BLOCK)
459                                     :   R0,R1,R2 DESTROYED
460                                     :
461                                     -
462
461 000640 012701 000106 ALLXCB: MOV    #X$LEN,R1    ;; GET SIZE OF X.25 CIRCUIT BLOCK
462 000644          CALL  $XALOC    ;; ALLOCATE CIRCUIT BLOCK
463 000650          BCS   50$        ;; BR IF UNSUCCESSFUL ALLOCATION
464 000652          MOV    @PSIPT,R2 ;; POINT TO PSI HOME BLOCK
465 000656          MOV    H$PTB(R2),-(SP) ;; GET UNMAPPED ADDRESS OF PORT TABLE
466 000662          CALL  $CEACX    ;; MAP TO PORT TABLE
467 000666          TST    (SP)+    ;; CLEAN UP STACK
468 000670          MOV    R0,@PRTADD ;; STORE UNMAPPED ADDRESS IN PORT TABLE
469 000674          CALL  INIXCB    ;; INITIALIZE X25 CIRCUIT BLOCK
470 000700          CALL  FNDHSH    ;; FIND HASH TABLE ENTRY AND SET UP
471 000704          BCS   50$        ;; BR IF NO ENTRY AVAILABLE
472 000706          MOV    PVCOWN,R0 ;; GET PROCESS OWNER NAME
473 000712          BEQ   40$        ;; IF EQ, NONE SPECIFIED
474 000714          CALL  FNDPDV    ;; GET PDV INDEX
475 000720          MOVB  R1,X$MOWN(R3) ;; STORE PDV INDEX
476 000724          40$: MOV    ZTIME,R1 ;; GET ADDRESS OF $ZTIME
477 000730          ADD    #2,R1    ;; $ZTIME+2
478 000734          MOV    (R1),X$TCLZ(R3) ;; STORE TIME SINCE LAST ZEROED
479 000740          50$: RETURN

```

\*\*FILE\*\*ID\*\*CFPRDT

K 10

|          |          |         |         |         |          |
|----------|----------|---------|---------|---------|----------|
| CCCCCCCC | FFFFFFFF | PPPPPPP | RRRRRRR | DDDDDDD | TTTTTTTT |
| CCCCCCCC | FFFFFFFF | PPPPPPP | RRRRRRR | DDDDDDD | TTTTTTTT |
| CC       | FF       | PP PP   | RR RR   | DD DD   | TT       |
| CC       | FF       | PP PP   | RR RR   | DD DD   | TT       |
| CC       | FF       | PP PP   | RR RR   | DD DD   | TT       |
| CC       | FF       | PP PP   | RR RR   | DD DD   | TT       |
| CC       | FFFFFFFF | PPPPPPP | RRRRRRR | DD DD   | TT       |
| CC       | FFFFFFFF | PPPPPPP | RRRRRRR | DD DD   | TT       |
| CC       | FF       | PP      | RR RR   | DD DD   | TT       |
| CC       | FF       | PP      | RR RR   | DD DD   | TT       |
| CC       | FF       | PP      | RR RR   | DD DD   | TT       |
| CC       | FF       | PP      | RR RR   | DD DD   | TT       |
| CC       | FF       | PP      | RR RR   | DD DD   | TT       |
| CCCCCCCC | FF       | PP      | RR RR   | DDDDDDD | TT       |
| CCCCCCCC | FF       | PP      | RR RR   | DDDDDDD | TT       |

....  
....  
....  
....

|            |          |          |
|------------|----------|----------|
| LL         | SSSSSSSS | TTTTTTTT |
| LL         | SSSSSSSS | TTTTTTTT |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SSSSSS   | TT       |
| LL         | SSSSSS   | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LL         | SS       | TT       |
| LLLLLLLLLL | SSSSSSSS | TT       |
| LLLLLLLLLL | SSSSSSSS | TT       |

CFPRDT CREATED BY MACRO ON 29-JUN-85 AT 00:19 PAGE 1 K 11  
 SYMBOL CROSS REFERENCE CREF 04.00

| SYMBOL  | VALUE       | REFERENCES                             |
|---------|-------------|--|
| CERR    | = ***** GX  | 8-103 8-104 8-105                      |
| CFGBF   | = ***** GX  | 9-128                                  |
| CFGSSZ  | = ***** GX  | 9-127                                  |
| CFLIN   | = ***** GX  | 8-103 8-104 8-105                      |
| CHKRDT  | = 000444 R  | 12-231 #13-293                         |
| COUNT   | = ***** GX  | *11-205 12-256                         |
| FMT10   | = ***** GX  | 8-103 8-104                            |
| FMT8    | = ***** GX  | 8-103                                  |
| FM.10   | = 000000    | #8-103 #8-104                          |
| FM.8    | = 000000    | #8-103                                 |
| H\$RDT  | = 000004    | 12-265 13-294 13-312                   |
| KSAR5   | = ***** GX  | 12-240 12-267                          |
| LNKEND  | = ***** GX  | 12-266                                 |
| NEXT    | = ***** GX  | *11-206 11-213                         |
| N\$SVCT | = *****     | 12-238 13-311                          |
| PCKBCD  | = ***** GX  | 12-230                                 |
| PSIPT   | = ***** GX  | 12-264 13-293                          |
| RDEND   | = 000212 R  | #12-226                                |
| RDNAM   | = 000052 R  | #11-178                                |
| RDTDF   | = 000000 R  | 9-124                                  |
| RDTE    | = 000006 R  | #7-94 11-201 11-206 12-228             |
| RDTEMX  | = 000006    | #6-82 11-179 12-252 13-306             |
| RDTEF   | = 000024 R  | #7-95 11-213                           |
| RDTEFNO | = 000166 R  | #11-213                                |
| RDTEP   | = 000025 R  | #7-96 12-229 12-258                    |
| RDTEST  | = 000130 R  | #11-200                                |
| RDTKW   | = 000000 RG | 9-126 #10-145                          |
| RDTLN   | = 000017    | #7-93 7-94 11-182 11-200 12-227 12-257 |
| RDTNAM  | = 000000 R  | #7-92 11-186 12-251 13-305             |
| RDTST   | = 000000 RG | #10-145                                |
| RTSPC   | = ***** GX  | 8-103 8-104 8-105                      |
| R\$DAL  | = 000013    | *12-256                                |
| R\$DTE  | = 000014    | 12-260                                 |
| R\$LEN  | = 000024    | 12-233                                 |
| R\$NAM  | = 000002    | 12-250 13-304                          |
| SPACE   | = 000040    | #6-83 11-183                           |
| STRNXT  | = ***** GX  | 11-207 11-215                          |
| SYNERR  | = ***** GX  | *9-129 *9-132                          |
| S\$BAS  | = *****     | 8-103 8-104 8-105 8-105                |
| \$ALPHA | = 000022    | #10-145                                |
| \$ANY   | = 000020    | #10-145                                |
| \$BLANK | = 000006    | #10-145                                |
| \$CEACX | = ***** GX  | 12-243 13-301                          |
| \$DIGIT | = 000024    | #10-145                                |
| \$DNUMB | = 000014    | #10-145                                |
| \$EOS   | = 000012    | #10-145                                |
| \$ERRLO | = 000036 R  | #8-103 11-195 12-275                   |
| \$ERRL1 | = 000066 R  | #8-104 11-220                          |
| \$ERRL2 | = 000122 R  | #8-105 12-276                          |
| \$ERRIT | = ***** GX  | 9-139                                  |
| \$EXIT  | = 000000    | #10-145                                |
| \$FAIL  | = 177777    | #10-145                                |

CFPX29      CREATED BY   MACRO   ON 29-JUN-85 AT 00:20      PAGE 2      K 12  
SYMBOL CROSS REFERENCE      CREF      04.00  
SYMBOL   VALUE      REFERENCES  
.TPARS   =   \*\*\*\*\*   GX      7-108



```

316 000436 103404          40$:   BCS      102$          ; BR IF UNABLE TO ALLOCATE CHANNELS LIST BLOCK
317 000440
318
319 ; ERRORS
320
321 000442          101$:   MSG$R   09          ; ILLEGAL CHANNEL RANGE VALUE
322 000450          102$:   MSG$R   08          ; RESOURCE ALLOCATION FAILURE
323
324
325          000001          .DSABL  LSB
                                .END
  
```

CTBIO - CETAB LOADING ROUTINES MACRO V05.03b Saturday 29-Jun-85 <sup>K 14</sup> 00:21 Page 9-2  
Symbol table

Work file reads: 0  
Work file writes: 0  
Size of work file: 10461 Words ( 41 Pages)  
Size of core pool: 14440 Words ( 55 Pages)  
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:09.05  
SY:CTBIO.V2,[132,134]CTBIO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CTBIO

55  
56  
57  
58  
59

.SBTTL MACRO DEFINITIONS  
:\*\*\*  
: LIBRARY MACROS  
:\*\*\*  
.MCALL DIR\$,Q10W\$

```

77                                     .SBTTL CREATE 'LOAD UIC STRING'
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
000000
000000 005046
000002 117716 000000G
000006 005046
000010 016700 000000G
000014 005200
000016 111016
000020 012700 000000'
000024 017702 000000G
000030 016202 000044
000034 010046
000036 010200
000040 016202 000000
000044 166200 000002
000050 016201 000010
000054
000060 066200 000006
000064 010001
000066 012600
000070 116220 000004
000074 116220 000005
000100 042701 177400
000104 001403
000106 005002
000110
000114 112720 000072
000120 112720 000133
000124 012601
000126 005002
000130
000134 112720 000054
000140 012601
000142 005002
000144
000150 112720 000135
000154 105010
000156 010001
000160 012700 000000'
000164 160001

```

```

+
$LDUIC - CREATE THE 'LOAD DEVICE' AND 'UIC' STRING
:
INPUTS:
$NTUIC = UIC USER CODE
:
OUTPUTS:
R0 = ADDRESS OF AN ASCIZ STRING OF THE FORM
      'DVNN:[NNN,NNN]', WHERE 'DVNN' IS
      THE LOAD DEVICE NMEMONIC.
R1 = LENGTH OF ASCIZ STRING (EXCLUDING ZERO BYTE)
R2 = DESTROYED
-
$LDUIC::
CLR      -(SP)                ;CLEAR A STACK ENTRY
MOVB     @NTUIC,(SP)          ;SAVE SECOND HALF OF UIC
CLR      -(SP)                ;CLEAR A SECOND ENTRY
MOVB     NTUIC,R0             ;GET $NTUIC ADDRESS
INC      R0                   ;$NTUIC+1
MOVB     (R0),(SP)            ;SAVE FIRST HALF OF UIC
MOVB     #UICBF,R0           ;POINT R0 AT TEMPORARY BUFFER
:
MOV      @TKTCB,R2            ;M-POINT AT TASK CONTROL BLOCK
MOV      T.LDV(R2),R2         ;M-POINT TO UCB OF LOAD DEVICE
MOV      R0,-(SP)             ;M-SAVE BUFFER POINTER
MOV      R2,R0                ;M-GET UCB ADDRESS
MOV      U.DCB(R2),R2         ;M-POINT AT DEVICE CONTROL BLOCK
SUB      D.UCB(R2),R0          ;M-COMPUTE RELATIVE UCB ADDRESS
MOV      D.UCBL(R2),R1        ;M-GET UCB LENGTH
CALL     $DIV                  ;M-COMPUTE RELATIVE UNIT NUMBER
ADD      D.UNIT(R2),R0        ;M-COMPUTE ABSOLUTE UNIT NUMBER
MOV      R0,R1                ;M-COPY UNIT NUMBER
MOV      (SP)+,R0             ;M-RESTORE BUFFER POINTER
MOVB     D.NAM(R2),(R0)+       ;M-FIRST LETTER OF DEVICE NAME
MOVB     D.NAM+1(R2),(R0)+     ;M-SECOND LETTER
:
BIC      #^C<377>,R1          ;UNIT ZERO ?
BEQ      10$                  ;IF EQ, YES - DON'T CONVERT
CLR      R2                   ;ALLOW ZERO SUPPRESSION
CALL     $CBOMG                ;PUT UNIT NUMBER IN STRING
MOVB     #'',(R0)+             ;PUT IN DEVICE IDENTIFIER
MOVB     #'L',(R0)+           ;PUT IN UIC START CHAR
MOV      (SP)+,R1              ;GET GROUP CODE FROM STACK
CLR      R2                   ;ALLOW ZERO SUPPRESSION
CALL     $CBOMG                ;PUT GROUP CODE IN UIC STRING
MOVB     #'',(R0)+             ;PUT IN UIC SEPARATOR
MOV      (SP)+,R1              ;GET USER CODE FROM STACK
CLR      R2                   ;ALLOW ZERO SUPPRESSION
CALL     $CBOMG                ;PUT USER CODE IN UIC STRING
MOVB     #'J',(R0)+           ;PUT IN UIC TERMINATOR
CLRB     (R0)                 ;MAKE INTO AN ASCIZ STRING
MOV      R0,R1                ;COPY ZERO BYTE ADDRESS
MOVB     #UICBF,R0            ;POINT AT BEGINNING OF BUFFER
SUB      R0,R1                ;GIVING LENGTH OF ASCIZ STRING

```

```

202      ; START OF CUG NUMBER
203
204
205      000136 012700 000004 CUNBST: MOV    #CUGNLN,R0      ; GET LENGTH OF CUG NUMBER
206      000142 012701 000006' MOV    #CUGNUM,R1      ; POINT TO START OF BUFFER
207      000146 105021 10$      10$: CLRB  (R1)+      ; INITIALIZE BUFFER
208      000150 005300      DEC    R0      ; MORE TO INITIALIZE?
209      000152 003375      BGT    10$      ; BR IF YES
210      000154 105067 000000G CLRB  COUNT      ; INITIALIZE COUNT FIELD
211      000160 012767 000006' 000000G MOV    #CUGNUM,NEXT ; START STORING AT BEGINNING OF BUFFER
212      000166      CALL  STRNXT      ; STORE NEXT CHARACTER
213      000172      RETURN
214      .E.ABL  LSB
215
216      ; PROCESS CUG NUMBER DIGIT
217
218      000174 026727 000000G 000011' CUGDIG: CMP    NEXT,#CUGEND ; IS ADDRESS TOO MANY CHARACTERS?
219      000202 101003      BHI    101$      ; BR IF YES
220      000204      CALL  STRNXT      ; STORE NEXT CHARACTER
221      000210      RETURN
222
223      ; ERRORS
224
225      000212      101$: MSG$R  N3      ; ILLEGAL CUG ADDRESS
226
227      ; END OF CUG ADDRESS
228
229      000220 126727 000000G 000001 CUNBEN: CMPB  COUNT,#1      ; ONE DIGIT ?
230      000226 001011      BNE    1$      ; IF NO - BRANCH
231      000230 116767 000006' 000007' MOVB  CUGNUM,CUGNUM+1 ; IF YES - INSERT LEADING ZERO
232      000236 112767 000060 000006' MOVB  #'0,CUGNUM ; ...
233      000244 105267 000000G      INCB  COUNT      ; ADJUST COUNT
234      000250 000422      BR     5$      ; CONTINUE
235
236      000252 126727 000000G 000003 1$: CMPB  COUNT,#3      ; THREE DIGITS ?
237      000260 001016      BNE    5$      ; IF NO - BRANCH
238      000262 012700 000010'      MOV    #CUGNUM+2,R0 ; IF YES - INSERT LEADING ZERO
239      000266 112010      MOVB  (R0)+,(R0) ; COPY THIRD DIGIT TO FOURTH POSITION
240      000270 012700 000007'      MOV    #CUGNUM+1,R0 ; COPY BUFFER ADDRESS
241      000274 112010      MOVB  (R0)+,(R0) ; COPY SECOND DIGIT TO THIRD POSITION
242      000276 116767 000006' 000007' MOVB  CUGNUM,CUGNUM+1 ; COPY FIRST DIGIT TO SECOND POSITION
243      000304 112767 000060 000006' MOVB  #'0,CUGNUM ; INSERT ZERO DIGIT
244      000312 105267 000000G      INCB  COUNT      ; ADJUST COUNT
245
246      000316 126727 000000G 000002 5$: CMPB  COUNT,#2      ; CUG ADDRESS MUST BE 2 OR 4 BYTES ONLY
247      000324 001004      BEQ    10$      ; BR IF VALID
248      000326 126727 000000G 000004      CMPB  COUNT,#4      ; IS IT VALID?
249      000334 001326      BNE    101$      ; BR IF NO
250      000336 004567 000000G      JSR    R5,$$SAVRG ; SAVE R3-R5
251      000342 012700 000002'      MOV    #<CUGNLN/2>,R0 ; GET LENGTH OF PACKED CUG ADDRESS
252      000346 012704 000006'      MOV    #CUGNUM,R4 ; POINT TO UNPACKED CUG ADDRESS
253      000352 012705 000012'      MOV    #CUGPCK,R5 ; POINT TO BUFFER TO STORE BCD ADDRESS
254      000356      CALL  PCKBCD ; PACK CUG ADDRESS IN BCD FORMAT
255      000362      RETURN
256      .DSABL  LSB
  
```

69  
70  
71 000000  
72  
73  
74  
75 000000  
76 000022  
77 000056  
78 000106  
79 000136  
80 000166  
81  
82  
83  
84  
85 000000

.SBTTL ERROR MESSAGES  
.PSECT DATA,D  
.NLIST BEX  
.ENABL LC  
NTLERS ,N2,10,CERR,RTSPC,CFLIN,<CUG name>  
NTLERS ,N8,10,CERR,RTSPC,CFLIN,<remote dte address>  
NTLERS ,N9,10,CERR,RTSPC,CFLIN,<low subaddress>  
NTLERS ,M0,10,CERR,RTSPC,CFLIN,<high subaddress>  
NTLERS ,M2,10,CERR,RTSPC,CFLIN,<call data value>  
NTLERS ,M3,10,CERR,RTSPC,CFLIN,<call data mask>  
.DSABL LC  
.EVEN  
.LIST BEX  
.PSECT

\*\*FILE\*\*ID\*\*CFPDST

|          |          |          |          |          |            |      |
|----------|----------|----------|----------|----------|------------|------|
| CCCCCCCC | FFFFFFFF | PPPPPPPP | DDDDDDDD | SSSSSSSS | TTTTTTTTTT |      |
| CCCCCCCC | FFFFFFFF | PPPPPPPP | DDDDDDDD | SSSSSSSS | TTTTTTTTTT |      |
| CC       | FF       | PP PP    | DD DD    | SS       | TT         |      |
| CC       | FF       | PP PP    | DD DD    | SS       | TT         |      |
| CC       | FF       | PP PP    | DD DD    | SS       | TT         |      |
| CC       | FF       | PP PP    | DD DD    | SS       | TT         |      |
| CC       | FFFFFFFF | PPPPPPPP | DD DD    | SSSSSS   | TT         |      |
| CC       | FFFFFFFF | PPPPPPPP | DD DD    | SSSSSS   | TT         |      |
| CC       | FF       | PP       | DD DD    | SS       | TT         |      |
| CC       | FF       | PP       | DD DD    | SS       | TT         |      |
| CC       | FF       | PP       | DD DD    | SS       | TT         |      |
| CC       | FF       | PP       | DD DD    | SS       | TT         | .... |
| CCCCCCCC | FF       | PP       | DDDDDDDD | SSSSSSSS | TT         | .... |
| CCCCCCCC | FF       | PP       | DDDDDDDD | SSSSSSSS | TT         | .... |

|            |          |            |
|------------|----------|------------|
| LL         | SSSSSSSS | TTTTTTTTTT |
| LL         | SSSSSSSS | TTTTTTTTTT |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SSSSSS   | TT         |
| LL         | SSSSSS   | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LL         | SS       | TT         |
| LLLLLLL -L | SSSSSSSS | TT         |
| LLLLLLLLLL | SSSSSSSS | TT         |

```

346                                     .SBTTL ALLDST - ALLOCATE DESTINATION DESCRIPTOR BLOCK
347
348                                     :+
349                                     ALLDST - ALLOCATE AND INITIALIZE DESTINATION DESCRIPTOR BLOCK
350
351                                     INPUTS:
352                                     R1 - NUMBER OF BYTES TO ALLOCATE FOR BLOCK
353
354                                     OUTPUTS:
355                                     CARRY CLEAR - ALLOCATION SUCCESSFUL
356                                     CARRY SET - UNSUCCESSFUL ALLOCATION
357
358
359
360 001104      ALLDST: SWSTK$ 100$      :: ENTER SYSTEM STATE
361 001110      CALL $XALOC      :: ALLOCATE BLOCK FROM NETWORK POOL
362 001114      BCC 10$          :: BR IF SUCCESSFUL ALLOCATION
363 001116      RETC R0          :: ELSE SET USER C-BIT
364 001130      BR 100$          :: AND EXIT
365 001132      10$: SAVMAP      :: SAVE CURRENT MAPPING
366 001136      MOV R0,-(SP)      :: SAVE UNMAPPED ADDRESS
367 001140      MOV R0,-(SP)      :: SET UNMAPPED ADDRESS FOR CONVERSION
368 001142      CALL $CEACK      :: CONVERT TO MAPPED ADDRESS
369 001146      MOV (SP)+,R0      :: RETRIEVE MAPPED ADDRESS
370 001150      MOV R0,R2        :: GET ADDRESS OF DESTINATION BLOCK
371 001152      15$: CLRB (R2)+  :: INITIALIZE BLOCK
372 001154      DEC R1           :: MORE TO INITIALIZE?
373 001156      BGT 15$          :: BR IF YES
374 001160      MOV #DSNAMX,R1    :: GET LENGTH OF DESTINATION NAME
375 001164      MOV R0,R2        :: GET ADDRESS OF DESTINATION BLOCK
376 001166      ADD #DSDST,R2     :: POINT TO DESTINATION NAME
377 001172      MOV #DSTNAM,R3    :: POINT TO SPECIFIED DESTINATION NAME
378 001176      20$: MOVB (R3)+,(R2)+ :: STORE DESTINATION NAME
379 001200      DEC R1           :: MORE TO STORE?
380 001202      BGT 20$          :: BR IF YES
381 001204      MOVB DSARCT,D$DTL(R0) :: STORE DIGIT COUNT FOR DTE ADDRESS
382 001212      MOVB DSCMCT,R3    :: GET CHARACTER COUNT FOR CALL DATA
383 001216      ASR R3           :: CONVERT TO BYTE COUNT
384 001220      ADC R3           ::
385 001222      MOVB R3,D$DATL(R0) :: STORE CALL DATA BYTE COUNT
386 001224      MOVB D$TPRI,D$PRI(R0) :: STORE PRIORITY
387 001234      MOVB D$TOBJ,D$OBJ(R0) :: STORE OBJECT NUMBER
388 001242      MOV D$TTSK,D$NAM(R0) :: STORE RAD50 TASK NAME
389 001250      MOV D$TTSK+2,D$NAM+2(R0) ::
390 001256      MOV D$ASLO,D$SLO(R0) :: STORE LOW END OF SUBADDRESS RANGE
391 001264      MOV D$ASHI,D$SHI(R0) :: STORE HIGH END OF DITTO
392 001272      MOV R0,R2        :: GET ADDRESS OF BLOCK
393 001274      ADD #D$CUGN,R2    :: POINT TO CUG NAME
394 001300      MOV #DSACUG,R3    :: POINT TO SPECIFIED NAME
395 001304      MOVB #CUNMMX,R1    :: GET LENGTH OF NAME
396 001310      35$: MOVB (R3)+,(R2)+ :: STORE CUG NAME
397 001312      DEC R1           :: MORE TO STORE?
398 001314      BGT 35$          :: BR IF YES
399 001316      ADD #D$VBL,R0     :: POINT TO START OF VARIABLE FIELDS
400 001322      MOV #D$TVAR,R1    :: POINT TO VARIABLE FIELD TABLE
401 001326      40$: MOV (R1)+,R2 :: GET ADDRESS OF COUNT FIELD
402 001330      BEQ 80$          :: BR IF END OF TABLE

```



```

LOCAL DATA
99          .SBTTL  LOCAL DATA
100
101          ;
102          ; LOCAL DATA
103          ;
104
105          .PSECT  DATA,D
106
107          .NLIST  BEX
108
109          ;
110          ; LOCAL DATA FOR DTE$DF MACRO
111          ;
112          000017      DTEALN = 15.          ; MAXIMUM NUMBER OF DIGITS IN DTE ADDRESS
113          000000      DTEADD: .BLKB  DTEALN      ; TEMPORARY STORAGE FOR UNPACKED DTE ADDRESS
114          000016'      DTEND = -1
115          000017      DTEPCK: .BLKB  8.          ; PACKED DTE ADDRESS
116          000006      DTNTNM: .BLKB  6.          ; MAX CHARACTERS IN NETWORK NAME
117          000027      DTNTNM: .BLKB  DTNMMX      ; NETWORK NAME
118          .EVEN
119

```

```

Symbol table

A$$CHK= 000000      HF$DLM= 000002      LF.TIM= 000010      L$NLR= 000104      N$$MCP= 000001
A$$CPS= 000000      HF$GWY= 000010      LF.UNL= 020000      L$NMAC 000076      N$$MLL= 000001
A$$PRI= 000000      HF$HQS= 000004      LF.X2P= 000000      L$NMA= 000074      N$$MDV= 000010
A$$TRP= 000000      HF$XDF= 000020      LINC TL= 000000    003 L$NMST 000003      N$$NCT= 000001
BIASX 001206R      HSHADD= 000000    003 L$NNRE 000106      N$$PEM= 000001
CERR = 000000 GX      HSHMN = 000040      L$NNRY 000046      PCKBCD= 000000 GX
CFGBF = 000000 GX      HSHMX = 001000      L$NRCA 000066      FSIPT = 000000 GX
CFGSZ = 000000 GX      HSHSZ = 000000 GX      L$NRFS 000100      P$$P45= 000000
CFLIN = 000000 GX      HSH TBL 000214R      L$NRPK 000056      P$$WRD= 000000
CHKDEV 001062R      H$CUG 000010      L$NRRE 000105      Q$$OPT= 000010
CHKDTE 000750R      H$DST 000012      L$NRST 000107      RTSPC = 000000 GX
CNTTIM 000264R      H$D29 000014      L$NTBY 000052      R$R0 = 000002
COUNT = 000000 GX      H$FLG 000000      L$NTCA 000070      R$R1 = 000004
CTIM = 000000 GX      H$GLEN 000104      L$NTFS 000102      R$R2 = 000006
C$SCKP= 000000      H$GLT 000044      L$NTPK 000062      R$R3 = 000010
C$SDRE= 000400      H$GNAM 000050      L$DMST 000002      R$R4 = 000012
C$SRSH= 177564      H$GNML= 000020      L$PCS 000112      R$R5 = 000014
DEVCTL= 000000 GX      H$GPT 000046      L$PLS 000110      R$SDER= 000000
DEVNAM= 000000 GX      H$HITS 000034      L$QCN 000122      R$SK11= 000001
DEVUNT= 000000 GX      H$HLEN 000044      L$QUEH 000124      R$SND= 000000
DE.OFF= 000000 GX      H$LBDA 000070      L$QUET 000130      R$STIM= 000000
DE.ON = 000000 GX      H$LBDN 000072      L$RTRY 000007      SF.ACT= 000200
DTEAD 000012R      003 H$LDTE 000002      L$SLN 000004      SF.ENA= 000100
DTEADD 000000R      002 H$LEN 000042      L$ST 000005      SF.LPB= 000004
DTEAEN 000140R      H$LDTS 000032      L$STCLZ 000044      SF.MFL= 000040
DTEALN= 000017      H$NETW 000024      L$TIM 000006      SF.PAC= 000020
DTEAST 000052R      H$NML = 000006      L$SASG= 000000      SF.REA= 000010
DTEDES= 000000 GX      H$NPT 000022      L$SDRV= 000000      SF.SER= 000001
DTEDF 000000R      003 H$PTB 000020      L$SP11= 000001      SF.SVC= 000002
DTEDIG 000114R      H$PVC 000006      L$ST11R= 000000      SF.UNL= 000040
DTEEND 000350R      H$RDT 000004      L.COST 000015      SLTMA = 000000 GX
DTEFLG= 000000 GX      H$RNW 000042      L.CTL 000012      SLTNM = 000000 GX
DTEKW 000000RG      004 H$SVC 000036      L.CVA 177776      SPACE = 000040
DTEND = 000016R      002 H$TRB 000016      L.DDM 000002      STRNXT= 000000 GX
DTENET 000310R      H$XAVL 000100      L.DDS 000004      SYNERR= 000000 GX
DTENW 000076R      003 H$XBIA 000074      L.DLC 000003      S$$WRG= 000000
DTEPCK 000017R      002 H$X29C 000040      L.DLM 000006      S$$YSZ= 007600
DTEST 000000RG      003 I$SRAR= 000000      L.DLS 000010      S.COST 000001
DTNMMX= 000006      I$SRDN= 000000      L.FLG 000000      S.FLG 000000
DTNTNM 000027R      002 K$AR5 = 000000 GX      L.KRBA 000016      S.LEN 000004
D$BUG= 177514      K$CNT= 177546      L.LEN = 000022      S.NMST 000002
D$ISK= 000000      K$CSR= 177546      L.MPF 000022      S.OWNR 000003
D$SL11= 000001      K$SLDC= 000000      L.NMST 000020      T$SKMG= 000000
D$SYNC= 000000      K$STPS= 000074      L.NSTA 000014      T$MIN= 000000
D$SYNM= 000000      LD$LP = 000000      L.DWNR 000021      V$CTR= 001000
END 000114R      003 LF.ACT= 100000      L.UNT 000013      X$DBT= 000000
E$XPR= 000000      LF.BRD= 000400      M$SCR= 000124      ZTIME = 000000 GX
FMT10 = 000000 GX      LF.BWT= 000007      M$CRX= 000000      ALPHA= 000022
FMT8 = 000000 GX      LF.ENA= 002000      M$FCS= 000000      SANY = 000020
FM.10 = 000000      LF.LPB= 001000      M$MGE= 000000      SBLANK= 000006
FM.8 = 000000      LF.MDC= 000100      M$NET= 000000      SCEACX= 000000 GX
FNDP DV= 000000 GX      LF.MFL= 004000      M$OVR= 000000      SDIGIT= 000024
F$SLVL= 000001      LF.MTP= 000020      NEXT = 000000 GX      SDNUMB= 000014
G$STP= 000000      LF.PAC= 000200      NUMDTE= 000020      SEDS = 000012
G$STSS= 000000      LF.RDY= 040000      N$ACC= 000001      ERRROD 000154R 002
G$STTK= 000000      LF.REA= 010000      N$SBUF= 000001      ERRROP 000214R 002
G$SWRD= 000000      LF.SER= 000040      L$NIRE 000072      N$LDV= 000001      ERROQ 000266R 002

```

```

113 .SBTTL ERROR MESSAGES
114
115 .ENABL LC
116
117 000014 NTLER$ ,P1,10,CERR,RTSPC,CFLIN,<occurrence of PSN$DF>
118 000052 NTLER$ ,P2,10,CERR,RTSPC,CFLIN,<network name>
119 000100 NTLER$ ,P3,10,CERR,RTSPC,CFLIN,<number of ports>
120 000130 NTLER$ ,P4,10,CERR,RTSPC,CFLIN,<flags word>
121 000154 NTLER$ ,P5,8,CERR,RTSPC,CFLIN,<PSI home block allocation failure>
122 000226 NTLER$ ,P6,8,CERR,RTSPC,CFLIN,<Port table allocation failure>
123 000274 NTLER$ ,D1,10,CERR,RTSPC,CFLIN,<low subaddresss value>
124 000332 NTLER$ ,D2,10,CERR,RTSPC,CFLIN,<high subaddress value>
125
126 .EVEN
127 .DSABL LC
128 .LIST BEX
129 000000 .PSECT

```

CFPPVC - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:18<sup>L 8</sup>  
Table of contents

|     |     |                                       |
|-----|-----|---------------------------------------|
| 5-  | 54  | MACRO DEFINITIONS                     |
| 6-  | 97  | LOCAL SYMBOL DEFINITIONS              |
| 7-  | 115 | LOCAL DATA                            |
| 8-  | 152 | ERROR MESSAGES                        |
| 9-  | 166 | LOOK FOR PVC\$DF MACRO                |
| 10- | 196 | TPARS STATE TABLES                    |
| 11- | 286 | PVC\$DF ACTION ROUTINES               |
| 14- | 447 | ALLXCB - ALLOCATE X25 CIRCUIT BLOCK   |
| 15- | 481 | CHKPVC - CHECK FOR VALID PVC          |
| 16- | 518 | FNDPRT - FIND FREE PORT ENTRY         |
| 17- | 555 | INIXCB - INITIALIZE X25 CIRCUIT BLOCK |
| 18- | 599 | FNDHSH - FIND HASH TABLE ENTRY        |
| 19- | 641 | BIASX - BIAS XPOOL ADDRESS FOR APR6   |
| 20- | 661 | PVDLM - CHECK FOR DLM PVC             |

```

481                                     .SBTTL  CHKPVC - CHECK FOR VALID PVC
482
483                                     *
484                                     :
485                                     : CHKPVC - CHECK FOR VALID PVC
486                                     :
487                                     : INPUTS:
488                                     :     NONE
489                                     :
490                                     : OUTPUTS:
491                                     :     CARRY CLEAR - SPECIFIED PVC IS UNIQUE
492                                     :     CARRY SET - SPECIFIED PVC IS NOT UNIQUE
493                                     :     R0 DESTROYED
494                                     :
495                                     -
496 000742 017700 0000006  CHKPVC: MOV    @PSIPT,R0      ; POINT TO PSI HOME BLOCK
497 000746 062700 000006   ADD    #H$PVC,R0      ; POINT TO PVC NAME BLOCK LISTHEAD
498
499 000752                                SWSTK$ 50$      ; ENTER SYSTEM STATE
500 000756                                SAVMAP      ; SAVE CURRENT MAPPING
501 000762 011000          10$: MOV    (R0),R0      ; GET ADDRESS OF NEXT BLOCK IN LIST
502 000764 001424          BEQ    40$              ; BR IF END OF LIST
503 000766 010046          MOV    R0,-(SP)         ; SET UP UNMAPPED ADDRESS
504 000770          CALL    $CEACX                ; CONVERT TO MAPPED ADDRESS
505 000774 012600          MOV    (SP)+,R0         ; RETRIEVE MAPPED ADDRESS
506 000776 012702 000006   MOV    #PVNAMX,R2      ; GET NUMBER OF CHARACTERS IN CIRCUIT ID
507 001002 012701 000024'   MOV    #PVCNAM,R1      ; POINT TO SPECIFIED PVC CIRCUIT ID
508 001006 010003          MOV    R0,R3           ; COPY PVC BLOCK POINTER
509 001010 062703 000002   ADD    #C$NAM,R3       ; POINT TO CIRCUIT ID FIELD
510 001014 122123          CMPB   (R1)+,(R3)+     ; IS THIS A UNIQUE CIRCUIT ID?
511 001016 001361          BNE    10$              ; BR IF YES, SO FAR - CONTINUE SEARCH
512 001020 005302          DEC    R2              ; MORE CHARACTERS TO CHECK?
513 001022 003374          BGT    20$              ; BR IF YES
514 001024          RETC    R0                    ; ELSE ERROR - SET USER C-BIT
515 001036          40$:  RESMAP      ; RESTORE PREVIOUS MAPPING
516 001042          50$:  RETURN

```

CFPRDT - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:19 <sup>L 10</sup>  
Table of contents

|     |     |                               |
|-----|-----|-------------------------------|
| 5-  | 55  | MACRO DEFINITIONS             |
| 6-  | 77  | LOCAL SYMBOL DEFINITIONS      |
| 7-  | 85  | LOCAL DATA                    |
| 8-  | 99  | ERROR MESSAGES                |
| 9-  | 112 | LOOK FOR RDT\$DF MACRO        |
| 10- | 141 | TPARS STATE TABLES            |
| 11- | 173 | RDT\$DF ACTION ROUTINES       |
| 13- | 278 | CHKRDT - CHECK FOR UNIQUE RDT |

CFPRDT      CREATED BY    MACRO    ON 29-JUN-85 AT 00:19      PAGE 2      L 11

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL    | VALUE       | REFERENCES    |
|-----------|-------------|---------------|
| \$GPRM    | = *****     | 10-145        |
| \$HEADR   | = ***** GX  | 12-238 13-311 |
| \$LAMDA   | = 000000    | #10-145       |
| \$NUMBR   | = 000002    | #10-145       |
| \$QRDT    | = 000000 RG | #9-124        |
| \$RAD50   | = 000016    | #10-145       |
| \$RONLY   | = *****     | 10-145 10-145 |
| \$SAVRG   | = ***** GX  | 12-226        |
| \$STRNG   | = 000004    | #10-145       |
| \$SUBXP   | = 000010    | #10-145       |
| \$XALOC   | = ***** GX  | 12-236        |
| \$\$\$FLG | = 177777    | #10-145       |
| \$\$\$KEY | = 177777    | #10-145       |
| .PSTCN    | = ***** GX  | 11-178        |
| .PSTPT    | = ***** GX  | 11-187        |
| .TPARS    | = ***** GX  | 9-130         |

CFPX29      CREATED BY    MACRO    ON 29-JUN-85 AT 00:20      PAGE 3      L 12  
 MACRO CROSS REFERENCE      CREF    04.00

MACRO NAME      REFERENCES

|         |        |        |        |        |        |        |        |        |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| CALL    | 7-108  | 9-183  |        |        |        |        |        |        |
| DBGTP\$ | #8-123 | #8-146 |        |        |        |        |        |        |
| EMSG\$R | #5-57  | 7-117  | 9-162  | 9-172  | 9-201  | 9-202  |        |        |
| ISTAT\$ | #5-57  | 8-123  |        |        |        |        |        |        |
| MTRAN\$ | #8-123 |        |        |        |        |        |        |        |
| NTLER\$ | #5-57  | 6-73   | 6-74   | 6-75   | 6-76   |        |        |        |
| PHBDF\$ | #5-57  | 5-60   |        |        |        |        |        |        |
| RETURN  | 7-112  | 9-160  | 9-170  | 9-199  |        |        |        |        |
| STATE\$ | #5-57  | 8-127  | #8-130 | #8-133 | #8-136 | #8-139 | #8-144 | #8-148 |
| TRAN\$  | #5-57  | #8-128 | #8-131 | #8-134 | #8-137 | #8-140 | #8-145 | #8-146 |
| X29DF\$ | #5-57  | 5-59   |        |        |        |        |        |        |



|                    |                   |                |                  |                      |
|--------------------|-------------------|----------------|------------------|----------------------|
| ALOCB = ***** GX   | G\$STSS= 000000   | L\$CTEN 000032 | L\$ASG= 000000   | X3PDF 000000R 003    |
| A\$CHK= 000000     | G\$STTK= 000000   | L\$CTIM 000040 | L\$DRV= 000000   | X3PKW 000000RG 004   |
| A\$CPS= 000000     | G\$SWRD= 000000   | L\$DTEA 000020 | L\$P11= 000001   | X3PST 000000RG 003   |
| A\$PRI= 000000     | I\$RAR= 000000    | L\$DTEL 000017 | L\$11R= 000000   | \$ALPHA= 000022      |
| A\$TRP= 000000     | I\$SRDN= 000000   | L\$GLEN 000134 | M\$CRB= 000124   | \$ANY = 000020       |
| BLKDEF 000060R     | K\$ARS = ***** GX | L\$LEN 000122  | M\$CRX= 000000   | \$BLANK= 000006      |
| BLKMAX 000102R     | K\$HIC 000004     | L\$LLCH 000016 | M\$FCS= 000000   | \$CEACX= ***** GX    |
| BLKSDZ= ***** GX   | K\$LEN 000006     | L\$LNK 000000  | M\$MGE= 000000   | \$DIGIT= 000024      |
| BLKSZM= ***** GX   | K\$LNK 000000     | L\$MCHN 000036 | M\$NET= 000000   | \$DNUMB= 000014      |
| CERR = ***** GX    | K\$LOC 000002     | L\$NETW 000114 | M\$DVR= 000000   | \$EDS = 000012       |
| CFGBF = ***** GX   | K\$CNT= 177546    | L\$NIRE 000072 | N\$ACC= 000001   | \$ERRD0= ***** GX    |
| CFGSZ = ***** GX   | K\$CSR= 177546    | L\$NLRE 000104 | N\$BUF= 000001   | \$ERRD1= ***** GX    |
| CFLIN = ***** GX   | K\$LDC= 000000    | L\$NMAC 000076 | N\$LDV= 000001   | \$ERRD2= ***** GX    |
| CHEND 000302R      | K\$TPS= 000074    | L\$NMAS 000074 | N\$MCP= 000001   | \$ERR08 000004R 002  |
| CHHI 000254R       | LD\$LP = 000000   | L\$NMST 000003 | N\$MLL= 000001   | \$ERRD9= ***** GX    |
| CHKBLK= ***** GX   | LN\$OFF= 000001   | L\$NNRE 000106 | N\$MDV= 000010   | \$ERRP7= ***** GX    |
| CHKWND= ***** GX   | LN\$ON = 000000   | L\$NRBY 000046 | N\$NCT= 000001   | \$ERRP8= ***** GX    |
| CHLD 000220R       | LN\$SHU= 000002   | L\$NRCA 000066 | N\$PEM= 000001   | \$ERRP9= ***** GX    |
| CHNDF 000036R 003  | LP\$EIP= 002000   | L\$NRFS 000100 | P\$P45= 000000   | \$ERR1T= ***** GX    |
| CHNLHI 000002R 002 | LP\$ENB= 004000   | L\$NRPK 000056 | P\$WRD= 000000   | \$EXIT = 000000      |
| CHNLLO 000000R 002 | LP\$PCT= 001400   | L\$NRRE 000105 | Q\$DPT= 000010   | \$FAIL = 177777      |
| CHNLNM= 000001     | LP\$TMR= 100000   | L\$NRST 000107 | RTSPC = ***** GX | \$HEADR= ***** GX    |
| CHNLMX= ***** GX   | LP\$UP = 010000   | L\$NTBY 000052 | R\$DER= 000000   | \$LAMBDA= 000000     |
| C\$CKP= 000000     | LP\$WTD= 020000   | L\$NTCA 000070 | R\$K11= 000001   | \$NUMBR= 000002      |
| C\$ORE= 000400     | LP\$WTS= 040000   | L\$NTFS 000102 | R\$SND= 000000   | \$QCHN 000006RG      |
| C\$SRSH= 177564    | L\$GDN= 000004    | L\$NTPK 000062 | R\$11M= 000000   | \$QX3P 000000RG      |
| DTEDES= ***** GX   | L\$OFF= 000000    | L\$DMST 000002 | SYNRR= ***** GX  | \$RAD50= 000016      |
| D\$BUG= 177514     | L\$STR= 000002    | L\$PCS 000112  | S\$WRG= 000000   | \$STRNG= 000004      |
| D\$ISK= 000000     | L\$SUP = 000003   | L\$PLS 000110  | S\$YSZ= 007600   | \$SUBXP= 000010      |
| D\$SL11= 000001    | L\$WT = 000001    | L\$QCN 000122  | T\$KMG= 000000   | \$SFLG= 177777       |
| D\$SYNC= 000000    | L\$ACHN 000012    | L\$QUEH 000124 | T\$MIN= 000000   | \$SKEY= 000001       |
| D\$YNM= 000000     | L\$APVC 000014    | L\$QUET 000130 | V\$CTR= 001000   | \$SSR = 000010       |
| END 000070R 003    | L\$ASVC 000010    | L\$RTRY 000007 | WNDDEF 000140R   | \$SSSTA= 000000      |
| E\$XPR= 000000     | L\$AUC 000042     | L\$SLN 000004  | WNDMAX 000162R   | \$SSTMP= 000007R 005 |
| FMT8 = ***** GX    | L\$CHLS 000034    | L\$ST 000005   | WNSZD= ***** GX  | .PNUMB= ***** GX     |
| FM.8 = 000000      | L\$CHTB 000030    | L\$TCLZ 000044 | WNSZM= ***** GX  | .PNUMH= ***** GX     |
| F\$LVL= 000001     | L\$CLEN= 000044   | L\$TIM 000006  | X\$DBT= 000000   | .TPARS= ***** GX     |
| G\$TPP= 000000     |                   |                |                  |                      |

. ABS. 000134 000 (RW,I,GBL,ABS,OVR)  
 000456 001 (RW,I,LCL,REL,CON)  
 DATA 000056 002 (RW,D,LCL,REL,CON)  
 \$STATE 000100 003 (RW,D,LCL,REL,CON)  
 \$KTAB 000004 004 (RW,D,LCL,REL,CON)  
 \$KSTR 000016 005 (RW,D,LCL,REL,CON)  
 Errors detected: 0

### \*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 13751 Words ( 54 Pages)  
 Size of core pool: 14440 Words ( 55 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:28.46  
 SY:CFPX3P.V2,[132,134]CFPX3P/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFPX3P

CTBIO      CREATED BY    MACRO    ON 29-JUN-85 AT 00:21      PAGE 1      L 14

SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL  | VALUE       | REFERENCES   |
|---------|-------------|--|
| DSKBUF  | 000000 R    | #6-126      8-190  |
| ERROR   | 000000 R    | #6-119      *7-148      7-158      7-161      *7-167      *7-169      *8-217      *8-219      *8-221 |
|         |             | *8-223      *8-225      *9-251   |
| ERR10   | 000012      | #5-94      8-219   |
| ERR12   | 000014      | #5-95  |
| ERR14   | 000016      | #5-96      8-223   |
| ERR16   | 000020      | #5-97      8-225   |
| ERR18   | 000022      | #5-98  |
| ERR2    | 000002      | #5-90      7-167   |
| ERR20   | 000024      | #5-99  |
| ERR22   | 000026      | #5-100   |
| ERR24   | 000030      | #5-101   |
| ERR26   | 000032      | #5-102   |
| ERR28   | 000034      | #5-103   |
| ERR30   | 000036      | #5-104   |
| ERR32   | 000040      | #5-105   |
| ERR34   | 000042      | #5-106      9-251  |
| ERR36   | 000044      | #5-107      3-221  |
| ERR4    | 000004      | #5-91  |
| ERR6    | 000006      | #5-92      7-169   |
| ERR8    | 000010      | #5-93      8-217   |
| LSBEXT  | 000352      | 8-200  |
| LSBFLG  | 000030      | 8-193  |
| LSBLDZ  | 000016      | 8-195  |
| LSBMXZ  | 000020      | 8-196  |
| PR.LBL  | 000076 R    | 7-154      #8-190  |
| PR.XFR  | 000272 R    | 7-156      #9-241  |
| R\$EIS  | = *****     | 8-204  |
| TS\$NHD | = 040000    | 8-193  |
| \$ALL16 | = ***** GX  | 8-208  |
| \$CLDEQ | = ***** GX  | 7-157  |
| \$CLIOS | = ***** GX  | 7-150      8-216      9-250  |
| \$CLOPE | = ***** GX  | 7-149  |
| \$CTBIO | = 000000 RG | #7-147   |
| \$IOSB  | = ***** GX  | 8-216      9-250   |
| \$ILEN  | = ***** GX  | *8-205   |
| \$LLEN  | = ***** GX  | 9-243  |
| \$READ  | = ***** GX  | 8-191  |
| \$RLBL  | = ***** GX  |  |

```

61                                     .SBTTL  LOCAL DATA
62                                     :***
63                                     : LOCAL DATA
64                                     :***
65
66 000000                                     .PSECT  DATA,D
67
68                                     :
69                                     : SAVE USER APR1
70                                     :
71 000000  UISAV:  .BLKW  1
72
73                                     :
74                                     : STATISTICS BLOCK
75                                     :
76 000002  $STAT:  .BLKW  5
77
78                                     :
79                                     : READ LOGICAL BLOCK DPB
80
81 000014      003      014  LBNDPB: .BYTE  3,12.      ; DIC, LENGTH
82 000016  000000G      .WORD  10,RLB      ; FUNCTION CODE
83 000020  000001      .WORD  1      ; LOGICAL UNIT
84 000022  000001      .WORD  1      ; EVENT FLAG
85 000024  000044      .WORD  $IOSB      ; I/O STATUS BLOCK
86 000026  000000      .WORD  0      ; AST
87 000030  000000  $LBUF: .WORD  0      ; BUFFER ADDRESS
88 000032  000000  $LLEN: .WORD  0      ; BUFFER LENGTH
89 000034  000000      .WORD  0      ; CARRIAGE CONTROL
90 000036      000000  $LBN: .BLKW  2      ; LOGICAL BLOCK NUMBER
91 000042  000000      .WORD  0
92
93                                     :
94                                     : I/O STATUS BLOCK
95                                     :
96 000044  $IOSB: .BLKW  2
97
98 000000                                     .PSECT

```

LDUIC - FIND THE LOAD UIC FOR N MACRO V05.03b Saturday 29-Jun-85 <sup>L 16</sup> 00:21 Page 6-1  
CREATE 'LOAD UIC STRING'

134 000166

135

136

000001

RETURN

;RETURN TO CALLER

.END

```

258      ;
259      ; CUG FLAGS BYTE
260      ;
261      ;
262      000364 005767 000000G      CUGF:  .ENAB!  LSB      ; LEGAL VALUE?
263      000370 001115      BNE      .PNUMH      ; BR IF NO
264      000372 105767 000001G      TSTB      .PNUMB+1      ; LEGAL VALUE?
265      000376 001112      BNE      101$      ; BR IF NO
266      000409 116767 000000G 000014'      MOVB      .PNUMB,CUGFLG      ; SAVE CUG FLAGS BYTE
267      000406      CALL      CHKCG      ; CHECK FOR VALID CUG NAME
268      000412 103507      BCS      102$      ; BR IF ERROR
269      000414 126727 000000G 000004      CMPB      COUNT,#4      ; IS CUG NUMBER 4 DIGITS?
270      000422 001003      BNE      10$      ; BR IF NO
271      000424 152767 000001 000014'      BISB      #GF$BUG,CUGFLG      ; FORCE BILATERAL CUG
272      ;
273      000432      10$: SWSTK$ 90$      ;: ENTER SYSTEM STATE
274      000436 012701 000016      30$: MOV      #G$LEN,R1      ;: GET NUMBER OF BYTES TO ALLOCATE
275      000442      CALL      $XALOC      ;: ALLOCATE BLOCK FROM NETWORK POOL
276      000446 103006      BCC      40$      ;: BR IF SUCCESS
277      000450      RETC      R0      ;: ELSE SET USER C-BIT
278      000462 000455      BR      80$      ;: AND EXIT
279      000464      40$: SAVMAP      ;: SAVE CURRENT MAPPING
280      000470 010005      MOV      R0,R5      ;: SAVE UNMAPPED ADDRESS
281      000472 010046      MOV      R0,-(SP)      ;: SET UP FOR CONVERSION
282      000474      CALL      $CEACX      ;: CONVERT TO MAPPED ADDRESS
283      000500 012600      MOV      (SP)+,R0      ;: RETRIEVE MAPPED ADDRESS
284      000502 010002      MOV      R0,R2      ;: GET ADDRESS OF CUG NAME BLOCK
285      000504 105022      45$: CLR      (R2)+      ;: INITIALIZE BLOCK
286      000506 005301      DEC      R1      ;: MORE TO INITIALIZE?
287      000510 003375      BGT      45$      ;: BR IF YES
288      000512 012701 000000G      MOV      #CUNMMX,R1      ;: GET LENGTH OF CUG NAME
289      000516 012702 000000'      MOV      #CUGNAM,R2      ;: POINT TO SPECIFIED CUG NAME
290      000522 010003      MOV      R0,R3      ;: POINT TO CUG NAME BLOCK
291      000524 062703 000002      ADD      #G$NAM,R3      ;: POINT TO CUG NAME
292      000530 112223      50$: MOV      (R2)+,(R3)+      ;: STORE CUG NAME IN CUG NAME BLOCK
293      000532 005301      DEC      R1      ;: MORE TO STORE?
294      000534 003375      BGT      50$      ;: BR IF YES
295      000536 012701 000002      MOV      #<CUGNLN/2>,R1      ;: GET LENGTH OF CUG NUMBER
296      000542 012702 000012'      MOV      #CUGPCK,R2      ;: POINT TO SPECIFIED CUG NUMBER (BCD)
297      000546 010003      MOV      R0,R3      ;: POINT TO CUG NAME BLOCK
298      000550 062703 000010      ADD      #G$CUG,R3      ;: POINT TO CUG NUMBER
299      000554 112223      60$: MOV      (R2)+,(R3)+      ;: STORE CUG NUMBER IN CUG NAME BLOCK
300      000556 005301      DEC      R1      ;: MORE TO STORE?
301      000560 003375      BGT      60$      ;: BR IF YES
302      000562 016760 000000G 000012      MOV      DTEDES,G$DTE(R0)      ;: STORE DTE ADDRESS
303      000570 116760 000014' 000014      MOV      CUGFLG,G$FLG(R0)      ;: STORE CUG FLAGS BYTE
304      000576 017700 000000G      MOV      @PSIPT,R0      ;: GET ADDRESS OF PSI HOME BLOCK
305      000602 062700 000010      ADD      #H$CUG,R0      ;: POINT TO CUG NAME BLOCK LISTHEAD
306      000606      CALL      LNKEND      ;: LINK AT END OF LIST
307      000612      70$: RESMAP      ;: RESTORE PREVIOUS MAPPING
308      000616      80$: RETURN      ;:
309      ;
310      000620 103407      90$: BCS      103$      ;: BR IF ERROR
311      000622      RETURN
312      ;
313      ; ERRORS
314      ;

```

```

87          .SBTTL  LOOK FOR DSA$DF MACRO
88
89
90          ;+
91          ;$QDSA - LOOK FOR DSA$DF MACRO
92          ;INPUTS:
93          ;      NONE
94
95          ;OUTPUTS:
96          ;      C-BIT=SUCCESS/FAILURE
97          ;      R3,R4,R5=DESTROYED
98          ;--
99          .ENABL  LSB
100
101 000000 012705 000000' $QDSA:: MOV #DSADF,R5 ; STATE TABLE ADDRESS
102 000004 000402          BR 10$
103 000006 012705 000130' $QDSC:: MOV #DSCDF,R5 ; STATE TABLE ADDRESS
104 000012 005001 10$: CLR R1 ; FULL KEYWORD MATCH LENGTH
105 000014 012702 000000' MOV #DSAKW,R2 ; KEYWORD TABLE ADDRESS
106 000020 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
107 000024 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
108 000030 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
109 000034 CALL .TPARS ; GO DO THE PARSE
110 000040 103003 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
111 000042 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
112 000046 001401 BEQ 101$ ; IF EQ, YES
113 000050 20$: RETURN
114
115          ;
116          ; ERROR CONDITION
117
118 000052 101$: MSG$R 11 ; SYNTAX ERROR
119          .DSABL  LSB
  
```

CFPDST - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 <sup>N 3</sup> 00:16  
Table of contents

|     |     |  |
|-----|-----|--|
| 5-  | 55  | MACRO DEFINITIONS                                      |
| 6-  | 76  | LOCAL SYMBOL DEFINITIONS                               |
| 7-  | 100 | ERROR MESSAGES   |
| 8-  | 117 | LOOK FOR DST\$DF MACRO                                 |
| 9-  | 147 | TPARS STATE TABLES                                     |
| 10- | 194 | DST\$DF ACTION ROUTINES                                |
| 13- | 278 | DSTEND - END OF DESTINATION DESCRIPTOR BLOCK MACROS    |
| 14- | 313 | CHKDSN - CHECK DESTINATION NAME                        |
| 15- | 346 | ALLDST - ALLOCATE DESTINATION DESCRIPTOR BLOCK         |
| 16- | 417 | DSTINS - INSERT DESTINATION DESCRIPTOR BLOCK INTO LIST |

|                   |               |             |                                   |
|-------------------|---------------|-------------|-----------------------------------|
| 403 001332 012104 | MOV           | (R1)+,R4    | :: GET ADDRESS OF SPECIFIED FIELD |
| 404 001334 112203 | MOVB          | (R2)+,R3    | :: GET LENGTH OF NEXT FIELD       |
| 405 001336 001773 | BEQ           | 40\$        | :: IF EQ, NOTHING TO STORE        |
| 406 001340 006203 | ASR           | R3          | :: CONVER, TO BYTE COUNT          |
| 407 001342 005503 | ADC           | R3          | ::                                |
| 408 001344 112420 | 70\$: MOVB    | (R4)+,(R0)+ | :: STORE SPECIFIED FIELD          |
| 409 001346 005303 | DEC           | R3          | :: MORE TO STORE?                 |
| 410 001350 003375 | BGT           | 70\$        | :: BR IF YES                      |
| 411 001352 000765 | BR            | 40\$        | :: GET NEXT FIELD TO STORE        |
| 412 001354 012601 | 80\$: MOV     | (SP)+,R1    | :: RETRIEVE UNMAPPED ADDRESS      |
| 413 001356        | CALL          | DSTINS      | :: INSERT BLOCK INTO LIST         |
| 414 001362        | RESMAP        |             | :: RESTORE PREVIOUS MAPPING       |
| 415 001366        | 100\$: RETURN |             |                                   |



ERROR MESSAGES

121  
122  
123  
124 000036  
125 000070  
126 000120  
127 000154  
128 000214  
129 000266  
130  
131  
132  
133 000000

.SBTTL ERROR MESSAGES

.ENABL LC  
NTLERS\$ ,03,10,CERR,RTSPC,CFLIN,<DTE address value>  
NTLERS\$ ,04,10,CERR,RTSPC,CFLIN,<hash table size>  
NTLERS\$ ,07,8,CERR,RTSPC,CFLIN,<Line not in system>  
NTLERS\$ ,00,10,CERR,RTSPC,CFLIN,<network name parameter>  
NTLERS\$ ,0P,8,CERR,RTSPC,CFLIN,<DTE descriptor allocation failure>  
NTLERS\$ ,0Q,8,CERR,RTSPC,CFLIN,<Too many DTE\$DFs>  
.DSABL LC  
.LIST BEX  
.EVEN  
.PSECT

CFPDTE - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:17 Page 18-2

Symbol table

|                   |                       |                   |                    |                      |
|-------------------|-----------------------|-------------------|--------------------|----------------------|
| \$ERRO3 000036R   | 002 \$FAIL = 177777   | \$RAD50= 000016   | \$\$\$FLG= 177777  | .PNUMB= ***** GX     |
| \$ERRO4 000070R   | 002 \$HEADR= ***** GX | \$SAVRG= ***** GX | \$\$\$KEY= 000002  | .PNUMH= ***** GX     |
| \$ERRO5= ***** GX | \$LAMDA= 00C000       | \$STRNG= 000004   | \$\$\$R = 000010   | .PSTCN= ***** GX     |
| \$ERRO7 000120R   | 002 \$NUMBER= 000002  | \$S'BXP= 000010   | \$\$\$STA= 000000  | .PSTPT= ***** GX     |
| \$ERRIT= ***** GX | \$QDTE 000000RG       | \$XALOC= ***** GX | \$\$\$TMP= 000012R | 005 .TPARS= ***** GX |
| \$EXIT = 000000   |                       |                   |                    |                      |

. ABS. 177776 000 (RW,I,GBL,ABS,OVR)  
 001226 001 (RW,I,LCL,REL,CON)  
 DATA 000320 002 (RW,D,LCL,REL,CON)  
 \$STATE 000154 003 (RW,D,LCL,REL,CON)  
 \$KTAB 000006 004 (RW,D,LCL,REL,CON)  
 \$KSTR 000016 005 (RW,D,LCL,REL,CON)  
 Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 16618 Words ( 65 Pages)  
 Size of core pool: 17608 Words ( 67 Pages)  
 Operating system: RSX-11M/PLUS

Elapsed time: 00:00:45.20  
 SY:CFPDTE.V2,[132,134]CFPDTE/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]CFPDTE

```

131                                     .SBTTL LOOK FOR PSN$DF MACRO
132
133                                     ;
134                                     ; $QPSN - LOOK FOR PSN$DF MACRO
135                                     ;
136                                     ; INPUTS:
137                                     ;     NONE
138                                     ;
139                                     ; OUTPUTS:
140                                     ;     C-BIT=SUCCESS/FAILURE
141                                     ;     R3,R4,R5=DESTROYED
142                                     ;
143 000000 012705 000000' $QPSN:: MOV #PSNDF,R5 ; STATE TABLE ADDRESS
144 000004 005001          CLR R1 ; FULL KEYWORD MATCH LENGTH
145 000006 012702 000000' MOV #PSNKW,R2 ; KEYWORD TABLE ADDRESS
146 000012 016703 000000G MOV CFGSZ,R3 ; RECORD LENGTH
147 000016 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
148 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
149 000026          CALL TPARS ; GO DO THE PARSE
150 000032 103003 BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
151 000034 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
152 000040 001401 BEQ 101$ ; IF EQ, YES
153 000042          20$: RETURN
154
155                                     ;
156                                     ; ERROR CONDITION
157                                     ;
158 000044          101$: MSG$R 1T ; SYNTAX ERROR

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52

.TITLE CFPPVC - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 30-JAN-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RX V1.0

.SBTTL FNDPRT - FIND FREE PORT ENTRY

```

FNDPRT - FIND FREE PORT ENTRY

INPUTS:
    PORTNO - TOTAL NUMBER OF PORTS

OUTPUTS:
    CARRY CLEAR - FREE PORT FOUND
    R1 - PORT NUMBER ALLOCATED
    PVCPR1 - PORT NUMBER
    PR1ADD - MAPPED ADDRESS OF ENTRY IN PORT TABLE
    CARRY SET - FREE PORT NOT FOUND

FNDPRT: SAVMAP          ;; SAVE CURRENT MAPPING
        MOV     @PSIPT,R0      ;; GET ADDRESS OF PSI HOME BLOCK
        MOV     H$PTB(R0),-(SP) ;; GET ADDRESS OF PORT TABLE
        CALL    $CEACX         ;; CONVERT TO MAPPED ADDRESS
        MOV     (SP)+,R0       ;; RETRIEVE MAPPED ADDRESS OF PORT TABLE
        CLR     R1             ;; PORT NUMBER COUNTER
10$:    TST      (R0)+          ;; SKIP FIRST ENTRY (NOT AVAILABLE)
        INC     R1             ;; INCREMENT PORT NUMBER
        TST      (R0)+          ;; IS THIS A FREE PORT?
        BNE     15$           ;; BR IF NO
        TST      -(R0)          ;; POINT TO PORT ENTRY
        MOV     R0,PR1ADD       ;; SAVE MAPPED ADDRESS OF ENTRY
        MOV     R1,PVCPR1       ;; SAVE PORT NUMBER ALLOCATED
        BR      20$            ;; AND EXIT
15$:    CMP     R1,PORTNO       ;; END OF LIST?
        BLO     10$            ;; BR IF NO
        SEC                     ;; INDICATE NUMBER OF PORTS EXCEEDED
20$:    RESMAP          ;; RESTORE PREVIOUS MAPPING
        RETURN

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

.TITLE CFPRDT - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

NTL - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 31-MAR-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/RSX V1.0

CFPRDT      CREATED BY    MACRO    ON 29-JUN-85 AT 00:19      PAGE 3      M 11

MACRO CROSS REFERENCE      CREF    04.00

| MACRO NAME | REFERENCES      |         |         |         |         |         |         |         |         |        |
|------------|-----------------|---------|---------|---------|---------|---------|---------|---------|---------|--------|
| CALL       | 9-130<br>13-301 | 11-207  | 11-215  | 12-230  | 12-231  | 12-235  | 12-236  | 12-243  | 12-266  | 13-296 |
| DBGTP\$    | #10-145         | #10-163 | #10-169 |         |         |         |         |         |         |        |
| EMSG\$R    | #5-60           | 9-139   | 11-195  | 11-220  | 12-275  | 12-276  |         |         |         |        |
| ISTAT\$    | #5-60           | 10-145  |         |         |         |         |         |         |         |        |
| MTRAN\$    | #10-145         |         |         |         |         |         |         |         |         |        |
| NTLER\$    | #5-60           | 8-103   | 8-104   | 8-105   |         |         |         |         |         |        |
| PHBDF\$    | #5-60           | 5-62    |         |         |         |         |         |         |         |        |
| RDTDF\$    | #5-60           | 5-63    |         |         |         |         |         |         |         |        |
| RESMAP     | #5-72           | 12-267  | 13-312  |         |         |         |         |         |         |        |
| RET\$      | #5-60           | 12-238  | 13-311  |         |         |         |         |         |         |        |
| RETURN     | 9-134           | 11-191  | 11-208  | 11-216  | 12-268  | 12-271  | 13-313  |         |         |        |
| SAVMAP     | #5-68           | 12-240  | 13-297  |         |         |         |         |         |         |        |
| STATE\$    | #5-60           | 10-149  | #10-152 | #10-155 | #10-158 | #10-161 | #10-167 | #10-171 |         |        |
| SWSTK\$    | 12-235          | 13-296  |         |         |         |         |         |         |         |        |
| TRANS      | #5-60           | #10-150 | #10-153 | #10-156 | #10-159 | #10-162 | #10-163 | #10-168 | #10-169 |        |

••FILE••ID••CFPX3P

N 12

```
CCCCCCCC FFFFFFFF PPPPPPP XX XX 333333 PPPPPPP
CCCCCCCC FFFFFFFF PPPPPPP XX XX 333333 PPPPPPP
CC FF PP PP XX XX 33 33 PP PP
CC FF PP PP XX XX 33 33 PP PP
CC FF PP PP XX XX 33 33 PP PP
CC FFFFFFFF PPPPPPP XX XX 33 33 PPPPPPP
CC FFFFFFFF PPPPPPP XX XX 33 33 PPPPPPP
CC FF PP XX XX 33 33 PP
CC FF PP XX XX 33 33 PP
CC FF PP XX XX 33 33 PP
CC FF PP XX XX 33 33 PP
CCCCCCCC FF PP XX XX 333333 PP
CCCCCCCC FF PP XX XX 333333 PP
```

```
LL SSSSSSS TTTTTTTTT
LL SSSSSSS TTTTTTTTT
LL SS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSS TT
LL SSSSS TT
LL SS TT
LL SS TT
LL SS TT
LL SSSSS TT
LLLLLLLLL SSSSSSS TT
LLLLLLLLL SSSSSSS TT
```



CFPX3P      CREATED BY MACRO ON 29-JUN-85 AT 00:20      PAGE 1      M 13  
 SYMBOL CROSS REFERENCE      CREF      04.00

| SYMBOL  | VALUE      | REFERENCES                    |
|---------|------------|-------------------------------|
| ALOCB   | = ***** GX | 12-288                        |
| BLKDEF  | 000060 R   | #11-208                       |
| BLKMAX  | 000102 R   | #11-220                       |
| BLKSZD  | = ***** GX | *11-210 11-222                |
| BLKSZM  | = ***** GX | *11-224                       |
| CERR    | = ***** GX | 7-99                          |
| CFGBF   | = ***** GX | 8-128                         |
| CFGSZ   | = ***** GX | 8-127                         |
| CFLIN   | = ***** GX | 7-99                          |
| CHEND   | 000302 R   | #12-285                       |
| CHHI    | 000254 R   | #12-278                       |
| CHKBLK  | = ***** GX | 11-208 11-220                 |
| CHKWND  | = ***** GX | 11-235 11-247                 |
| CHLO    | 000220 R   | #12-265                       |
| CHNDF   | 000036 R   | 8-123                         |
| CHNLHI  | 000002 R   | *12-272 *12-284 12-293 12-306 |
| CHNLLD  | 000000 R   | *12-271 12-292 12-307         |
| CHNLMN  | = 00       | #6-82                         |
| CHNLMX  | = ***** GX | 12-269 12-282                 |
| DTEDES  | = ***** GX | 12-295                        |
| FMT8    | = ***** GX | 7-99                          |
| FM.8    | = 000000   | #7-99                         |
| KSARS   | = ***** GX | 12-294 12-313                 |
| K\$HIC  | 000004     | *12-293                       |
| K\$LEN  | 000006     | 12-285                        |
| K\$LOC  | 000002     | *12-292                       |
| L\$CHLS | 000034     | 12-299                        |
| L\$MCHN | 000036     | 12-311                        |
| N\$VCT  | = *****    | 12-290                        |
| RTSPC   | = ***** GX | 7-99                          |
| SYNERR  | = ***** GX | *8-129 *8-132                 |
| S\$BAS  | = *****    | 7-99                          |
| WNDDEF  | 000140 R   | #11-235                       |
| WNDMAX  | 000162 R   | #11-247                       |
| WNSZD   | = ***** GX | *11-237 11-249                |
| WNSZM   | = ***** GX | *11-251                       |
| X3PDF   | 000000 R   | 8-121                         |
| X3PKW   | 000000 RG  | 8-126 #9-147                  |
| X3PST   | 000000 RG  | #9-147                        |
| \$ALPHA | = 000022   | #9-147                        |
| \$ANY   | = 000020   | #9-147                        |
| \$BLANK | = 000006   | #9-147                        |
| \$CEACX | = ***** GX | 12-296                        |
| \$DIGIT | = 000024   | #9-147                        |
| \$DNUMB | = 000014   | #9-147                        |
| \$EOS   | = 000012   | #9-147                        |
| \$ERRC0 | = ***** GX | 11-256                        |
| \$ERR01 | = ***** GX | 11-230                        |
| \$ERR02 | = ***** GX | 11-257                        |
| \$ERR08 | 000004 R   | #7-99 12-322                  |
| \$ERR09 | = ***** GX | 12-321                        |
| \$ERRP7 | = ***** GX | 11-215                        |

CTBIO      CREATED BY    MACRO    ON 29-JUN-85 AT 00:21      PAGE 2      M 14

MACRO CROSS REFERENCE      CREF    04.00

MACRO NAME      REFERENCES

|         |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|
| ASL\$   | #5-71 | 8-204 |       |       |       |       |       |
| CALL    | 7-149 | 7-154 | 7-156 | 7-157 | 8-191 | 8-208 | 9-243 |
| HWDDF\$ | #5-56 | 5-61  |       |       |       |       |       |
| LBLDF\$ | #5-56 | 5-62  |       |       |       |       |       |
| RETURN  | 7-162 | 8-212 | 8-227 | 9-245 | 9-252 |       |       |
| SWSTK\$ | #5-56 |       |       |       |       |       |       |

```

100                                     .SBTTL  $CLOPE - OPEN A FILE
101                                     :+
102                                     $CLOPE - USE GCL TO OPEN A CONTIGUOUS FILE
103                                     :
104                                     INPUTS:
105                                     R1=FILE SPEC ADDRESS
106                                     R2=FILE SPEC LENGTH
107                                     :
108                                     OUTPUTS:
109                                     R0=STATISTICS BLOCK ADDRESS
110                                     $LBN=LABEL BLOCK LOGICAL BLOCK NUMBER
111                                     :
112 000000                                     $CLOPE::
113 000000 012700 000002'                   MOV    #STAT,R0           ; GET STATISTICS WHILE OPENING
114 000004                                     CALL   $CLOST          ; USE GCL TO OPEN THE FILE
115 000010 012700 000004'                   MOV    #STAT+2,R0        ; GET STATISTICS BLOCK ADDRESS
116 000014 011067 000040'                   MOV    (R0),$LBN+2      ; COPY LABEL BLOCK LBN
117 000020 014067 000036'                   MOV    -(R0),$LBN        ; ...
118 000024                                     RETURN

```

|                |                  |                  |                |                |
|----------------|------------------|------------------|----------------|----------------|
| ASSCHK= 000000 | LSSP11= 000001   | S2.WRB= 000002   | TS.STP= 001000 | UC.ATT= 000010 |
| ASSCPS= 000000 | LSS11R= 000000   | S3.ACR= 000001   | TSKMG= 000000  | UC.KIL= 000004 |
| ASSPRI= 000000 | MSSCRB= 000124   | S3.CTC= 000004   | TSMIN= 000000  | UC.LGH= 000003 |
| ASSTRP= 000000 | MSSCRX= 000000   | S3.FDX= 000200   | T.ACTL= 000052 | UC.NPR= 000100 |
| CSSCKP= 000000 | MSSFCS= 000000   | S3.ICE= 001000   | T.ASTL= 000016 | UC.PWF= 000020 |
| CSSORE= 000400 | MSSMGE= 000000   | S3.MHE= 000400   | T.ATT= 000062  | UC.QUE= 000040 |
| CSSRSH= 177564 | MSSNET= 000000   | S3.NEC= 000020   | T.CPCB= 000004 | UD.UNS= 000000 |
| DV.CCL= 000002 | MSSOVR= 000000   | S3.PPT= 020000   | T.DPRI= 000040 | UD.160= 000004 |
| DV.COM= 020000 | NTUIC = ***** GX | S3.PTH= 004000   | T.EFLG= 000022 | UD.200= 000001 |
| DV.DIR= 000010 | NSSACC= 000001   | S3.RAL= 000010   | T.IOC= 000003  | UD.556= 000002 |
| DV.EXT= 000400 | NSSBUF= 000001   | S3.RES= 010000   | T.LBN= 000041  | UD.625= 000005 |
| DV.F11= 040000 | NSSLDV= 000001   | S3.RUB= 040000   | T.LDV= 000044  | UD.8K = 000006 |
| DV.ISP= 002000 | NSSMCP= 000001   | S3.TAB= 000002   | T.LNK= 000000  | UD.800= 000003 |
| DV.MBC= 000400 | NSSMLL= 000001   | S3.TME= 002000   | T.MXSZ= 000050 | UICBF= 000000R |
| DV.MNT= 100000 | NSSMOV= 000010   | S3.TSV= 000040   | T.NAM= 000006  | UM.CLI= 000036 |
| DV.MSD= 000100 | NSSNET= 000001   | S3.8BC= 000100   | T.OFF= 000066  | UM.DSB= 000200 |
| DV.OSP= 004000 | NSS'CT= 000001   | S4.ABD= 100000   | T.PCB= 000046  | UM.NBR= 000400 |
| DV.PSE= 010000 | PSSP45= 000000   | S4.ANI= 000400   | T.PRI= 000002  | UM.OVR= 000001 |
| DV.REC= 000001 | PSSWRD= 000000   | S4.AVO= 001000   | T.RCVL= 000012 | UM.ABO= 000001 |
| DV.SDI= 000020 | QSSOPT= 000010   | S4.BLK= 002000   | T.RRFL= 000072 | US.BSP= 000002 |
| DV.SQD= 000040 | RSSDER= 000000   | S4.DE= 004000    | T.SAST= 000054 | US.BSY= 000200 |
| DV.SWL= 001000 | RSSK11= 000001   | S4.DLD= 000100   | T.SRCT= 000071 | US.FOR= 000040 |
| DV.TTY= 000004 | RSSSND= 000000   | S4.EDT= 010000   | T.STAT= 000032 | US.FRK= 000002 |
| DV.UMD= 000200 | RSS11M= 000000   | S4.HFF= 000020   | T.ST2= 000034  | US.KPF= 000001 |
| DSSBUG= 177514 | SSSWRG= 000000   | S4.HFI= 000007   | T.ST3= 000036  | US.LAB= 000004 |
| DSSISK= 000000 | SSSYSZ= 007600   | S4.HHT= 000040   | T.TCBL= 000030 | US.MDE= 000002 |
| DSSL11= 000001 | ST.DEC= 001000   | S4.HSY= 000200   | T.TID= 000057  | US.MDM= 000020 |
| DSSYNC= 000000 | ST.DPR= 000400   | S4.RGS= 020000   | T.TKSZ= 000060 | US.MNT= 000100 |
| DSSYNM= 000000 | ST.DSI= 004000   | S4.SFC= 040000   | T.UCB= 000026  | US.OFL= 000001 |
| D.DSP= 000012  | ST.ESC= 000002   | S4.VFC= 000010   | T2.ABO= 000100 | US.PUB= 000004 |
| D.LNK= 000000  | ST.1BF= 002000   | S5.ABP= 100000   | T2.AST= 100000 | US.PWF= 000010 |
| D.MSK= 000014  | ST.1BY= 000200   | S5.BCC= 020000   | T2.CAF= 000400 | US.RED= 000002 |
| D.NAM= 000004  | ST.1OBY= 000100  | S5.DAO= 040000   | T2.CHK= 020000 | US.SHR= 000001 |
| D.PCB= 000034  | ST.PTH= 000010   | S5.HPC= 000014   | T2.CKD= 010000 | US.SPU= 000002 |
| D.UCB= 000002  | ST.RES= 010000   | S5.HPD= 000020   | T2.DST= 040000 | US.UMD= 000010 |
| D.UCBL= 000010 | ST.RNE= 000020   | S5.JTF= 000100   | T2.FXD= 002000 | US.VV = 000001 |
| D.UNIT= 000006 | ST.RNF= 020000   | S5.OXF= 000040   | T2.HLT= 000200 | US.WCK= 000010 |
| D.VCAN= 000002 | ST.RSP= 000004   | S5.RPO= 002000   | T2.REX= 001000 | UU.ABO= 000400 |
| D.VDEB= 177776 | ST.RST= 000001   | S5.SWI= 000001   | T2.SEF= 004000 | UU.ATN= 000100 |
| D.VINI= 000000 | ST.TNE= 040000   | S5.TMM= 000002   | T2.SPN= 000004 | UU.AVN= 000004 |
| D.VOUT= 000004 | ST.TSY= 000040   | S5.VER= 010000   | T2.STP= 000020 | UU.GUS= 000010 |
| D.VPWF= 000006 | ST.USI= 100000   | S5.XOF= 000004   | T2.WFR= 000001 | UU.IOS= 002000 |
| FSSXPR= 000000 | S2.BEL= 020000   | S5.XON= 000010   | T3.ACP= 100000 | UU.ONL= 000020 |
| FSSLVL= 000001 | S2.BRO= 000020   | S6.EIO= 000400   | T3.CAL= 000100 | UU.RCT= 000002 |
| GSSTPP= 000000 | S2.CR = 002000   | S6.RDI= 100000   | T3.CLI= 001000 | UU.RDY= 000200 |
| GSSSTS= 000000 | S2.CTO= 040000   | S6.RLI= 001000   | T3.GFL= 000010 | UU.SER= 000001 |
| GSSTTK= 000000 | S2.CTS= 100000   | TKTCB = ***** GX | T3.MCR= 004000 | UU.SID= 001000 |
| GSSWRD= 000000 | S2.ELF= 001000   | TS.BLK= 171700   | T3.NET= 000020 | UU.SPC= 000040 |
| ISSRAR= 000000 | S2.FLF= 000400   | TS.CKP= 000200   | T3.NSD= 000200 | U.ACB = 000062 |
| ISSRDN= 000000 | S2.IR3= 000200   | TS.CKR= 000100   | T3.PMD= 040000 | U.ACP = 000032 |
| KSSCNT= 177546 | S2.OBF= 004000   | TS.EXE= 100000   | T3.PRV= 010000 | U.ADMA= 000066 |
| KSSCSR= 177546 | S2.ORG= 000100   | TS.HLD= 002000   | T3.REM= 020000 | U.AFLG= 000064 |
| KSSLDC= 000000 | S2.PCU= 010000   | TS.MSG= 020000   | T3.RDV= 000040 | U.ATT= 000022  |
| KSSTPS= 000074 | S2.RCU= 000001   | TS.NRP= 010000   | T3.RST= 000400 | U.BPKT= 000044 |
| LDLPL = 000000 | S2.SRO= 000040   | TS.OUT= 000400   | T3.SLV= 002000 | U.BUF= 000024  |
| LSSASG= 000000 | S2.WAL= 000010   | TS.RDN= 040000   | T3.SWS= 000002 | U.CBF = 000032 |
| LSSDRV= 000000 | S2.WRA= 000006   | TS.RUN= 004000   | UC.ALG= 000200 | U.CLI= 177772  |

315 000624  
316 000632  
317 000640  
318

101\$: EMSG\$R P4  
102\$: EMSG\$R N2  
103\$: EMSG\$R M1  
.DSABL LSB

; ILLEGAL FLAGS BYTE VALUE  
; ILLEGAL CUG NAME  
; CUG NAME BLOCK ALLOCATION FAILURE

```

121 .SBTTL TPARS STATE TABLES
122
123 : TPARS STATE TABLES
124 :
125 ISTAT$ DSAST,DSAKW
126 :
127 DSA$DF
128 :
129 STATES$ DSADF
130 TRANS$ %DSA$DF%,1,SYNERR
131
132 STATES$
133 TRANS$ !END,$EXIT
134 TRANS$ <'>,SLO
135 TRANS$ $DIGIT,DARDNX,DARDST ; START OF REMOTE DTE ADDRESS
136 TRANS$ <'>,,DARDST
137
138 STATES$ DARDNX
139 TRANS$ $DIGIT,DARDNX,DARDNO
140 TRANS$ <'>,DARDNX,DARDNO
141 TRANS$ !END,$EXIT,DARDND
142 TRANS$ <'>,,DARDND ; END OF REMOTE DTE ADDRESS
143
144 STATES$ SLO
145 TRANS$ <'>,SHI
146 TRANS$ $NUMBR,,DASSLO ; LOW END OF SUBADDRESS RANGE
147
148 STATES$
149 TRANS$ !END,$EXIT
150 TRANS$ <'>
151
152 STATES$ SHI
153 TRANS$ <'>,DACUG
154 TRANS$ $NUMBR,,DASSHI ; HIGH END OF SUBADDRESS RANGE
155
156 STATES$
157 TRANS$ !END,$EXIT
158 TRANS$ <'>
159
160 STATES$ DACUM,END,DACUNM ; CUG ADDRESS
161 TRANS$

```

.TITLE CFPDST - INITIAL CONFIG FILE SCAN ACTION ROUTINES  
.IDENT /V05.00/

COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

MODULE DESCRIPTION:

Ntl - CONFIG FILE ACTION ROUTINES

DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING

IDENT HISTORY:

- 1.00 31-MAR-81  
DECNET-11M/S V3.0  
DECNET-11M-PLUS V1.0
- 3.00 16-APR-82  
DECNET-11M V3.1  
DECNET-11M-PLUS V1.1
- 4.00 07-NOV-83  
DECNET-11M V4.0  
DECNET-11M-PLUS V2.0
- 5.00 22-JUL-85  
DECnet-11M/S V4.2  
DECnet-11M-Plus V3.0  
DECnet-Micro/Rsx V1.0

```

417 .SBTTL DSTINS - INSERT DESTINATION DESCRIPTOR BLOCK INTO LIST
418
419 ;+
420 ;
421 ; DSTINS - INSERT DESTINATION DESCRIPTOR BLOCK INTO LIST
422 ;
423 ; INPUTS:
424 ; R1 - UNMAPPED ADDRESS OF BLOCK TO LINK
425 ; DSTPRI - SPECIFIED PRIORITY
426 ;
427 ; OUTPUTS:
428 ; BLOCK IS LINKED INTO LIST BY PRIORITY
429 ; R2,R3,R4 DESTROYED
430 ;
431 001370 017702 000000G DSTINS: MOV @PSIPT,R2 ;; GET ADDRESS OF PS1 HOME BLOCK
432 001374 062702 000012 ADD #HSDST,R2 ;; POINT TO X25 LISTHEAD
433 001400 132767 000000G 000000G BITB #DS.X29,DSFLG ;; IS THIS AN X29 DESCRIPTOR BLOCK?
434 001406 001402 BEQ 10$ ;; NO, X25 IT IS
435 001410 062702 000002 ADD #HSD29-HSDST,R2 ;; ELSE, POINT TO X29 LISTHEAD
436 001414 010203 10$: MOV R2,R3 ;; MAKE EXTRA "MAPPED" COPY
437
438 001416 010204 20$: MOV R2,R4 ;; SAVE UNMAPPED ADDRESS OF PREVIOUS
439 001420 011302 MOV (R3),R2 ;; GET UNMAPPED ADDRESS OF NEXT
440 001422 001410 BEQ 40$ ;; BR IF END OF LIST - LINK HERE
441 001424 010246 MOV R2,-(SP) ;; SET ADDRESS FOR CONVERSION
442 001426 CALL $CEACK ;; CONVERT TO MAPPED ADDRESS
443 001432 012603 MOV (SP)+,R3 ;; RETRIEVE MAPPED ADDRESS
444 001434 126763 000000G 000002 CMPB DSTPRI,D$PRI(R3) ;; LINK BLOCK BEFORE THIS ONE?
445 001442 101765 BLOS 20$ ;; BR IF NO
446
447 001444 010146 40$: MOV R1,-(SP) ;; GET UNMAPPED ADDR OF BLOCK TO LINK
448 001446 010446 MOV R4,-(SP) ;; GET ADDRESS OF PREVIOUS BLOCK
449 001450 CALL $LINKX ;; LINK BLOCK INTO LIST
450 001454 RETURN ;;
451
452 000001 .END

```



```

135                                     .SBTTL LOOK FOR DTE$DF MACRO
136
137                                     ;+
138                                     ; $DTE - LOOK FOR DTE$DF MACRO
139                                     ;
140                                     ; INPUTS:
141                                     ;     NONE
142                                     ;
143                                     ; OUTPUTS:
144                                     ;     C-BIT=SUCCESS/FAILURE
145                                     ;     R3,R4,R5=DESTROYED
146                                     ; -
147 000000 012705 000000' $DTE:: MOV #DTE$DF,R5 ; STATE TABLE ADDRESS
148 000004 005001          CLR R1 ; FULL KEYWORD MATCH LENGTH
149 000006 012702 000000' MOV #DTE$KW,R2 ; KEYWORD TABLE ADDRESS
150 000012 016703 000000G MOV CFSZ,R3 ; RECORD LENGTH
151 000016 012704 000000G MOV #CFGBF,R4 ; RECORD BUFFER ADDRESS
152 000022 005067 000000G CLR SYNERR ; CLEAR SYNTAX ERROR FLAG
153 000026          CALL TPARS ; GO DO THE PARSE
154 000032 103003          BCC 20$ ; IF CC, FOUND WHAT WE WERE LOOKING FOR
155 000034 005367 000000G DEC SYNERR ; DID SYNTAX ERROR OCCUR?
156 000040 001401          BEQ 101$ ; IF EQ, YES
157 000042          20$: RETURN
158
159                                     ;
160                                     ; ERROR CONDITION
161                                     ;
162 000044          101$: MSG$R 1T ; SYNTAX ERROR

```

CFPDTE      CREATED BY    MACRO    ON 29-JUN-85 AT 00:17      PAGE 1      N 6  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL | VALUE      | REFERENCES                               |
|--------|------------|--|
| BIASX  | 001206 R   | 15-403 #18-530                           |
| CERR   | = ***** GX | 8-124 8-125 8-126 8-127 8-128 8-129      |
| CFGBF  | = ***** GX | 9-151                                    |
| CFGSZ  | = ***** GX | 9-150                                    |
| CFLIN  | = ***** GX | 8-124 8-125 8-126 8-127 8-128 8-129      |
| CHKDEV | 001062 R   | 15-344 #17-484                           |
| CHKDTE | 000750 R   | 15-340 #16-439                           |
| CNTTIM | 000264 R   | #13-307                                  |
| COUNT  | = ***** GX | *12-249 15-375                           |
| CTIM   | = ***** GX | *13-309 15-391                           |
| DE.ON  | = ***** GX | 15-396                                   |
| DTEADD | 000000 R   | #7-113 12-245 12-250 12-271              |
| DTEAEN | 000140 R   | #12-269                                  |
| DTEALN | = 000017 R | #7-112 7-113 12-244 12-270 15-370 16-451 |
| DTEAST | 000052 R   | #12-243                                  |
| DTEDES | = ***** GX | *15-359 15-408                           |
| DTEDF  | 000000 R   | 9-147                                    |
| DTEDIG | 000114 R   | #12-257                                  |
| DTEEND | 000350 R   | 14-329 #15-340                           |
| DTEFLG | = ***** GX | *12-243 15-396                           |
| DTEKW  | 000000 RG  | 9-149 #10-138                            |
| DTEND  | = 000016 R | #7-114 12-257                            |
| DTENET | 000310 R   | #14-321                                  |
| DTEPCK | 000017 R   | #7-115 12-272 15-371 16-454              |
| DTEST  | 000000 RG  | #10-168                                  |
| DTNMMX | = 000006 R | #7-116 7-117 12-278 14-322 15-379        |
| DTNTNM | 000327 R   | #7-117 12-277 14-325 15-330              |
| FMT10  | = ***** GX | 8-124 8-125 8-127                        |
| FMT8   | = ***** GX | 8-126 8-128 8-129                        |
| FM.10  | = 000000   | #8-124 #8-125 #8-127                     |
| FM.8   | = 000000   | #8-126 #8-128 #8-129                     |
| FNDPDV | = ***** GX | 17-490                                   |
| HSHADD | = ***** GX | *15-402                                  |
| HSHMN  | = 000040   | #6-85 13-290                             |
| HSHMX  | = 001000   | #6-86 13-294                             |
| HSHSZ  | = ***** GX | *13-297 15-347 15-405                    |
| HSHIBL | 000214 R   | #13-287                                  |
| HSLDTE | 000002     | 15-407 16-440                            |
| HNETW  | 000024     | 12-276                                   |
| KSAR5  | = ***** GX | 15-358 15-410 16-443 16-460              |
| LF.ACT | = 100000   | #5-67                                    |
| LF.BRO | = 000400   | #5-67                                    |
| LF.BWT | = 000007   | #5-67                                    |
| LF.ENA | = 002000   | #5-67                                    |
| LF.LPB | = 001000   | #5-67                                    |
| LF.MDC | = 000100   | #5-67                                    |
| LF.MFL | = 004000   | #5-67                                    |
| LF.MTP | = 000020   | #5-67 17-509                             |
| LF.PAC | = 000200   | #5-67                                    |
| LF.RDY | = 040000   | #5-67                                    |
| LF.REA | = 010000   | #5-67                                    |
| LF.SER | = 000040   | #5-67                                    |

```

160 .SBTTL TPARS STATE TABLES
161 ;
162 ; TPARS STATE TABLES
163 ;
164 ISTAT$ PSNST,PSNKW
165
166 ; PSN$DF
167 ;
168 ;
169 STATES$ PSNDF
170 TRANS$ %PSN$DF%,...1,SYNERR
171
172 STATES$
173 TRANS$ $STRNG,,PSNM ; NETWORK NAME
174
175 STATES$
176 TRANS$ <'>
177
178 STATES$
179 TRANS$ $NUMBR,,PORT ; NUMBER OF PORTS SUPPORTED
180
181 STATES$
182 TRANS$ !END,$EXIT,PSNEND
183 TRANS$ <'>
184
185 STATES$
186 TRANS$ <'>,PSLO
187 TRANS$ $NUMBR,,PSNFG ; FLAGS
188
189 STATES$
190 TRANS$ !END,$EXIT,PSNEND
191 TRANS$ <'>
192
193 STATES$ PSLO
194 TRANS$ <'>,PSHI
195 TRANS$ $NUMBR,,PSNSLO ; LOW END OF SUBADDRESS RANGE
196
197 STATES$
198 TRANS$ !END,$EXIT,PSNEND
199 TRANS$ <'>
200
201 STATES$ PSHI
202 TRANS$ $NUMBR,END,PSNSHI ; HIGH END OF SUBADDRESS RANGE
203
204 STATES$ END ; END OF STRING SUB-EXPRESSION
205 TRANS$ $EOS,$EXIT
206 TRANS$ <'>,$EXIT
207
208 STATES$

```

```

54      .SBTTL  MACRO DEFINITIONS
55
56      ;;
57      ;; LIBRARY MACROS
58      ;;
59      .MCALL  EMMSG$,SAVRG,RESRG,RET$,PVCDF$,XCBD$,
60      .MCALL  PHBDF$,NTLR$,STATE$,TRAN$,ISTAT$
61
62      ;; LIBRARY SYMBOLS
63      ;;
64      PHBDF$      ; PS1 HOME BLOCK OFFSETS
65      PVCDF$      ; PVC NAME BLOCK OFFSETS
66      XCBD$       ; X25 CIRCUIT CONTROL BLOCKS
67
68      ;; LOCAL MACRO DEFINITIONS
69      ;;
70      .MACRO  SAVMAP
71      MOV     @KSAR5,-(SP)      ; SAVE APR 5
72      .ENDM
73
74      .MACRO  RESMAP
75      MOV     (SP)+,@KSAR5     ; RESTORE APR5
76      .ENDM
77
78      ;; SET UP APR5 MAPPING
79      ;;
80      .MACRO  MAP      BIAS
81      MOV     BIAS,@KSAR5     ; SET UP APR5 MAPPING
82      .ENDM
83
84      ;; SET UP APR5 BIAS
85      ;;
86      .MACRO  BIAS      REG
87      .IF DF M$$MGE
88      BIC     #16000,REG      ; CLEAR CURRENT MAPPING BITS
89      .IF DF R$$11M
90      BIS     #12000,REG      ; SET UP APR5 MAPPING
91      .IFF
92      BIS     #40000,REG      ; SET UP APR2 MAPPING
93      .ENDC
94      .ENDC
95      .ENDM

```

```

555 .SBTTL INIXCB - INITIALIZE X25 CIRCUIT BLOCK
556
557
558
559 INIXCB - INITIALIZE X25 CIRCUIT BLOCK
560
561 INPUTS:
562 R0 - UNMAPPED ADDRESS OF CIRCUIT BLOCK
563
564 OUTPUTS:
565 R3 - MAPPED ADDRESS OF X25 CIRCUIT BLOCK
566 X25 CIRCUIT BLOCK INITIALIZED ACCORDING TO VALUES
567 SPECIFIED IN PVC$DF
568 R0,R1,R2 DESTROYED
569
570
571 INIXCB: MOV R0, -(SP) ;; GET UNMAPPED ADDRESS OF CIRCUIT BLOCK
572 CALL $CEACX ;; MAP TO CIRCUIT BLOCK
573 MOV (SP)+, R3 ;; RETRIEVE MAPPED CIRCUIT BLOCK ADDRESS
574 MOV #X$LEN, R1 ;; GET SIZE OF CIRCUIT BLOCK
575 MOV R3, R2 ;; GET CIRCUIT BLOCK ADDRESS
576 10$: CLRB (R2)+ ;; INITIALIZE CIRCUIT BLOCK TO ZEROES
577 DEC R1 ;; MORE TO INITIALIZE?
578 BGT 10$ BR IF YES
579 MOV R3, R0 ;; COPY VIRTUAL ADDRESS OF XLB
580 CALL BIASX ;; BIAS CORRECTLY FOR APR6
581 MOV PVCWSZ, X$WSZ(R3) ;; STORE WINDOW SIZE
582 MOV PVCPSZ, X$PKSZ(R3) ;; STORE PACKET SIZE
583 MOV PVCCTM, X$CTIM(R3) ;; STORE COUNTER TIMER
584 MOV DTEDES, X$DTE(R3) ;; STORE DTE DESCRIPTOR ADDRESS
585 MOV PVCPRN, X$PRT(R3) ;; STORE PORT NUMBER
586 MOV PVCLCN, X$LCN(R3) ;; STORE LOGICAL CHANNEL NUMBER
587 MOV R0, X$TXQ+2(R3) ;; INITIALISE TRANSMISSION QUEUE
588 ADD #X$TXQ, X$TXQ+2(R3) ;;
589 MOV R0, X$RXQ+2(R3) ;; INITIALISE RECEIVED DATA QUEUE
590 ADD #X$RXQ, X$RXQ+2(R3) ;;
591 MOV R0, X$WAQ+2(R3) ;; INITIALISE ACKNOWLEDGEMENT QUEUE
592 ADD #X$WAQ, X$WAQ+2(R3) ;;
593 MOV R0, X$ABQ+2(R3) ;; INITIALISE ABORTED DATA QUEUE
594 ADD #X$ABQ, X$ABQ+2(R3) ;;
595 MOV SP$P4, X$ST(R3) ;; SET PORT IN DATA TRANSFER STATE
596 MOV #XT$PVC, X$TYP(R3) ;; SET CIRCUIT TYP4 AS PVC
597 RETURN
  
```

55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67  
 68  
 69  
 70  
 71  
 72  
 73  
 74  
 75

.SBTTL MACRO DEFINITIONS

:: LIBRARY MACROS  
 ::

.MCALL RDTDF\$,RETC,EMSG\$R,PHBDF\$,NTLER\$,STATE\$,ISTAT\$,TRANS

PHBDF\$ ; PSI HOME BLOCK OFFSETS  
 RDTDF\$ ; REMOTE DTE NAME BLOCK OFFSETS

:: LOCAL MACRO DEFINITIONS  
 ::

.MACRO SAVMAP  
 MOV @KSAR5,-(SP) ; SAVE APR 5  
 .ENDM

.MACRO RESMAP  
 MOV (SP)+,@KSAR5 ; RESTORE APR5  
 .ENDM

```

CCCCCCCC  FFFFFFFF  PPPPPPP  XX    XX    222222  999999
CCCCCCCC  FFFFFFFF  PPPPPPP  XX    XX    222222  999999
CC         FF        PP        PP    PP    22    22  99    99
CC         FF        PP        PP    PP    XX    XX  22    22  99    99
CC         FF        PP        PP    PP    XX    XX  22    22  99    99
CC         FF        PP        PP    PP    XX    XX  22    22  99    99
CC         FFFFFFFF  PPPPPPP  XX    XX    22    22  99999999
CC         FFFFFFFF  PPPPPPP  XX    XX    22    22  99999999
CC         FF        PP        XX    XX    22    22  99    99
CC         FF        PP        XX    XX    22    22  99    99
CC         FF        PP        XX    XX    22    22  99    99
CC         FF        PP        XX    XX    22    22  99    99
CCCCCCCC  FF        PP        XX    XX    222222222  999999
CCCCCCCC  FF        PP        XX    XX    222222222  999999

```

```

LL          SSSSSSSS  TTTTTTTTTT
LL          SSSSSSSS  TTTTTTTTTT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LL          SSSSSS    TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SS        TT
LL          SSSSSS    TT
LLLLLLLLLL SSSSSSSS  TT
LLLLLLLLLL SSSSSSSS  TT

```

CFPX3P - INITIAL CONFIG FILE SC MACRO V05.03b Saturday 29-Jun-85 00:20 <sup>N 12</sup>  
Table of contents

|     |     |                          |
|-----|-----|--------------------------|
| 5-  | 55  | MACRO DEFINITIONS        |
| 6-  | 77  | LOCAL SYMBOL DEFINITIONS |
| 7-  | 84  | LOCAL DATA               |
| 8-  | 107 | LOOK FOR ***\$DF MACRO   |
| 9-  | 143 | TPARS STATE TABLES       |
| 10- | 176 | CHN\$DF STATE TABLE      |
| 11- | 203 | X3P\$DF ACTION ROUTINES  |
| 12- | 259 | CHN\$DF ACTION ROUTINES  |



CFPX3P      CREATED BY MACRO ON 29-JUN-85 AT 00:20      PAGE 2      N 13  
 SYMBOL CROSS REFERENCE      CREF    04.00

| SYMBOL    | VALUE      | REFERENCES         |
|-----------|------------|--------------------|
| \$ERRP8   | = ***** GX | 11-229             |
| \$ERRP9   | = ***** GX | 11-242             |
| \$ERR1T   | = ***** GX | 8-139              |
| \$EXIT    | = 000000   | #9-147             |
| \$FAIL    | = 177777   | #9-147             |
| \$GPRM    | = *****    | 9-147              |
| \$HEADR   | = ***** GX | 12-290             |
| \$LAMDA   | = 000000   | #9-147             |
| \$NUMBR   | = 000002   | #9-147             |
| \$QCHN    | 000006 RG  | #8-123             |
| \$QX3P    | 000000 RG  | #8-121             |
| \$RAD50   | = 000016   | #9-147             |
| \$RONLY   | = *****    | 9-147      9-147   |
| \$STRNG   | = 000004   | #9-147             |
| \$SUBXP   | = 000010   | #9-147             |
| \$\$\$FLG | = 177777   | #9-147             |
| \$\$\$KEY | = 177777   | #9-147             |
| .PNUMB    | = ***** GX | 12-267      12-280 |
| .PNUMH    | = ***** GX | 12-265      12-278 |
| .TPARS    | = ***** GX | 8-130              |

••FILE••ID••ELTIME

N 14

```
EEEEEEEEEE LL      TTTTTTTTTT 111111 MM MM EEEEEEEEEE
EEEEEEEEEE LL      TTTTTTTTTT 111111 MM MM EEEEEEEEEE
EE          LL      TT          11 MM MM EE
EE          LL      TT          11 MMMM MMMM EE
EE          LL      TT          11 MMMM MMMM EE
EE          LL      TT          11 MM MM MM EE
EE          LL      TT          11 MM MM MM EE
EEEEEEEEEE LL      TT          11 MM MM EEEEEEEEEE
EEEEEEEEEE LL      TT          11 MM MM EEEEEEEEEE
EE          LL      TT          11 MM MM EE
EE          LL      TT          11 MM MM EE
EE          LL      TT          11 MM MM EE
EE          LL      TT          11 MM MM EE
EEEEEEEEEE LLLLLLLLLL TT      111111 MM MM EEEEEEEEEE
EEEEEEEEEE LLLLLLLLLL TT      111111 MM MM EEEEEEEEEE
```

```
LL          SSSSSSSS TTTTTTTTTT
LL          SSSSSSSS TTTTTTTTTT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LL          SSSSSS  TT
LL          SSSSSS  TT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LL          SS      TT
LLLLLLLLLL SSSSSSSS TT
LLLLLLLLLL SSSSSSSS TT
```

```

120 .SBTTL $RLBL - READ LABEL BLOCK
121
122 * $RLBL - READ LABEL BLOCK
123
124 : INPUTS:
125 : R0=BUFFER ADDRESS
126
127 : OUTPUTS:
128 : C-BIT=SUCCESS/FAILURE
129 : $LBN=FIRST TEXT BLOCK LOGICAL BLOCK NUMBER
130 : $LLEN=512.
131
132 $RLBL::
133 000026 012767 001000 000032' MOV #512, $LLEN ; SET BUFFER LENGTH
134 000034 CALL $READ ; READ LABEL BLOCK
135 000040 006016 ROR (SP) ; SAVE C-BIT
136
137 .IF DF R$11D!1$SAS
138
139 BIT #1,$NHD, L$FLG(R0) ; IS THERE A HEADER PRESENT?
140 BEQ 10$ ; YES ... HEADER OFFSET IS CORRECT
141 MOV #1, L$BHRB(R0) ; NO ... USE CORRECT OFFSET FOR HEADER
142 10$:
143
144 .ENDC
145
146 000042 066067 000000G 000040' ADD L$BHRB(R0), $LBN+2 ; SET LBN OF FIRST TEXT BLOCK
147 000050 005567 000036' ADC $LBN ;
148 000054 006316 ASL (SP) ; RESTORE C-BIT
149 000056 RETURN

```

